

# Studie zu Fachapplikation Justiz

## Best-of-Breed-Strategie

**Auftraggeber**    Dr. Jens Piesbergen, Direktor  
**Autor**    Ehrenreich AG: Daniel Hasselwander, Nicolai Ehrenreich, Dr. Hans-Ruedi Troxler  
**Datum**    31.10.2025  
**Version**    V1.0

## Historie

| Version | Datum      | Beschreibung       | Namen               |
|---------|------------|--------------------|---------------------|
| V1.0    | 31.10.2025 | Version 1 erstellt | Daniel Hasselwander |

## Inhaltsverzeichnis

|       |   |    |
|-------|---|----|
| 1     | Ausgangslage .....  | 5  |
| 1.1   | Allgemeines bezüglich Fachapplikationen .....   | 5  |
| 1.1.1 | Paradigmenwechsel von Papier zur elektronischen Aktenführung - ganzheitlich entlang der Justizkette ..... | 5  |
| 1.1.2 | Grosser Anpassungsbedarf im Hinblick auf ERV .....  | 6  |
| 1.1.3 | Generell sind die bestehenden Lösungen veraltet und müssen erneuert werden. ....                          | 7  |
| 1.1.4 | Sehr oft monolithischer Aufbau .....  | 7  |
| 1.1.5 | Wenig flexible Architektur .....  | 8  |
| 1.1.6 | Anspruchsvoller Unterhalt und eingeschränktes Potential zur Weiterentwicklung .....                       | 9  |
| 1.2   | Nutzerorganisationen / Anwenderseite .....  | 10 |
| 1.3   | Lieferantensituation.....   | 10 |
| 1.4   | Fazit und Folgerungen .....   | 11 |
| 1.4.1 | Notwendigkeit neuer Anbieter mit modernem Ansatz.....   | 12 |
| 1.4.2 | Wissenstransfer und Domänenkompetenz als Schlüsselfaktor .....  | 12 |
| 1.4.3 | Risiko der Lieferantenabhängigkeit minimieren .....   | 12 |
| 1.4.4 | Best-of-Breed-Ansatz als erfolgversprechende Strategie.....   | 12 |
| 2     | Zielsetzung Studie .....  | 12 |
| 2.1   | Technische Definition einer langfristigen Lösung .....  | 12 |
| 2.2   | Machbarkeitsanalyse einer Best-of-Breed-Strategie .....   | 13 |
| 2.3   | Chancen- und Risikoanalyse .....  | 13 |
| 2.4   | Methodik .....  | 13 |
| 3     | Situation bei den Fachapplikationen und Lieferanten.....  | 13 |
| 3.1   | Übersicht Fachapplikationen im Justizbereich .....  | 14 |
| 3.1.1 | Staatsanwaltschaften.....   | 14 |
| 3.1.2 | Gerichte .....  | 14 |
| 3.1.3 | Justizvollzugsbehörden.....   | 15 |
| 3.2   | Fazit zur Situation bei den Fachapplikationen .....   | 16 |
| 4     | Bedürfnisse und Herausforderungen auf Anwenderseite .....   | 17 |

|       |  |    |
|-------|--|----|
| 4.1.1 | Anwender erwarten stabile, benutzerfreundliche Systeme, die ihre täglichen Arbeitsprozesse effektiv unterstützen .....                             | 17 |
| 4.1.2 | Organisationen bringen Anforderungen mit, die jene der User ergänzen und Aspekte wie Governance, Sicherheit und Nachhaltigkeit sicherstellen ..... | 17 |
| 4.1.3 | Durch den Einbezug der Justizbehörden werden mögliche Bedürfnisse zu konkreten Anforderungen .....   | 18 |
| 4.1.4 | Funktionsgruppen Landkarte .....   | 19 |
| 4.2   | Hohe Anforderungen an eine Fachapplikation der Zukunft .....   | 20 |
| 5     | Zielarchitektur .....  | 20 |
| 5.1   | Grundsätze der künftigen Architektur .....   | 21 |
| 5.1.1 | Anforderungen zeigen Grenzen monolithischer Architekturen in aktuellen Applikationen auf. ....   | 21 |
| 5.1.2 | Modularisierung als strategische Antwort .....   | 22 |
| 5.1.3 | Die richtige Segmentierung als Schlüssel.....  | 23 |
| 5.1.4 | Ereignisorientierung als architektonisches Ordnungsprinzip .....   | 24 |
| 5.1.5 | Event Sourcing: Revisionssicherheit durch vollständige Ereignisprotokollierung .....   | 25 |
| 5.1.6 | Dokumentation und Standardisierung als Voraussetzung für Interoperabilität .....   | 26 |
| 5.1.7 | Zusammenfassung .....  | 27 |
| 5.2   | Schematischer Aufbau der Zielarchitektur einer modernen Fachapplikation.....   | 27 |
| 5.2.1 | High-Level-Architektur und Komponentenüberblick.....   | 27 |
| 5.2.2 | Zusammenspiel der Komponenten am Beispielprozess Strafbefehl versenden.....  | 29 |
| 5.2.3 | Kernapplikation - Kommunikation und Datenhaltung .....   | 31 |
| 5.2.4 | Eventhub: Rolle und Anforderungen .....  | 35 |
| 5.2.5 | Breeds und Adapter im Detail .....   | 37 |
| 5.2.6 | Justizaktenapplikation – JAA .....   | 40 |
| 5.2.7 | Abschluss des Kapitels Zielarchitektur Fachapplikation .....   | 41 |
| 6     | Betriebsmodell in einer modularen, ereignisgesteuerten Justizarchitektur.....  | 42 |
| 6.1   | Ausgangslage und Herausforderungen .....   | 42 |
| 6.2   | Notwendigkeit differenzierter Betriebsmodelle.....   | 42 |
| 6.2.1 | Modell 1: Inhouse-Modell .....   | 43 |
| 6.2.2 | Modell 2: Kooperationsmodell mehrerer Organisationen.....  | 44 |
| 6.2.3 | Modell 3: Betrieb mehrerer Mandanten bei einem externen Dienstleister .....  | 45 |

|       |   |    |
|-------|---|----|
| 6.2.4 | Technische Betriebsgrundlagen .....   | 46 |
| 6.3   | Verknüpfung von Betriebsmodell und Betriebstools: Ein Auswahlprozess .....                    | 48 |
| 6.3.1 | Ausgangspunkt: Zwei Dimensionen .....   | 48 |
| 6.3.2 | Kriterien und Entscheidungsprozess zur Auswahl des passenden Betriebsmodells .....            | 48 |
| 7     | Generalunternehmermodell – Steuerung, Integration und Effizienz in der Projektumsetzung ..... | 50 |
| 7.1   | Anforderungen an potenzielle Generalunternehmer .....   | 50 |
| 7.1.1 | Technische Anforderungen an den Generalunternehmer .....                                      | 51 |
| 7.1.2 | Nicht-technische Anforderungen an einen Generalunternehmer .....                              | 52 |
| 7.2   | Auslegeordnung Generalunternehmer .....   | 52 |
| 8     | Fazit .....   | 53 |
| 8.1   | Zusammenfassung .....   | 53 |
| 8.2   | Machbarkeit der vorgeschlagenen Lösung .....  | 54 |
| 8.3   | Nächste Schritte .....  | 55 |

## 1 Ausgangslage

### 1.1 Allgemeines bezüglich Fachapplikationen

#### 1.1.1 Paradigmenwechsel von Papier zur elektronischen Aktenführung - ganzheitlich entlang der Justizkette

Die Schweizer Justiz steht vor einem fundamentalen Wandel, der über technologische Neuerungen hinausgeht: dem strukturellen Übergang von papiergestützten Verfahren zu einer vollständig elektronischen Aktenführung (eAkte) über alle Stufen des Straf-, Zivil- und Verwaltungsverfahren hinweg. Dieser Paradigmenwechsel betrifft sämtliche Akteure – von der Polizei über die Staatsanwaltschaft, die Gerichte, den Justizvollzug, weitere Verwaltungsstellen bis hin zur Anwaltschaft sowie weiteren professionellen Dritten wie Gutachter und Opferhilfestellen.

#### Ganzheitliche Aktenführung – von der Strafverfolgung bis zur Urteilsumsetzung

Im Strafverfahren beginnt die Aktenführung bereits mit der Feststellung des Sachverhalts und Erhebung von Beweismitteln durch die Polizei, geht über in die Strafuntersuchung durch die Staatsanwaltschaft, wird durch die Gerichte in der Hauptverhandlung fortgeführt und endet im Justizvollzug mit der Umsetzung von Sanktionen. Die eAkte muss diesen gesamten Informationsfluss durchgängig digital abbilden – ohne Medienbrüche, mit klar definierten Zuständigkeiten, Zugriffskontrollen und Protokollierung.

Auch im Zivilverfahren – zum Beispiel bei Scheidungen, Unterhaltsfragen oder Erbstreitigkeiten – ist die Einführung der eAkte entscheidend, um Fristen, Eingaben, Beweismittel und Protokolle elektronisch und nachvollziehbar zu verwalten. Besonders wichtig ist hier die strukturierte Abbildung komplexer Parteikontellationen sowie die Anbindung von zivilrechtlich tätigen Anwälten und Gerichtsschreibende.

#### Staatsanwaltschaften und Gerichte: Taktgeber des Aktenprozesses

Staatsanwaltschaften und Gerichte sind gleichermassen die organisatorischen Zentren des Justizverfahrens – sie bündeln Eingaben, führen die Verhandlungen und erlassen Entscheide. Die Einführung der eAkte in den Gerichten bringt tiefgreifende Veränderungen in den täglichen Arbeitsabläufen:

- Digitale Prozessführung: Verhandlungen mit digitaler Akteneinsicht, Kommentierfunktionen, Beweismittelmanagement
- Aktenkoordination: Gleichzeitiger Zugriff verschiedener Mitarbeitender auf dieselbe elektronische Akte
- Beschleunigung der Bearbeitungsprozesse: Vermeidung physischer (Papier)-Umläufe, bessere Fristenkontrolle, strukturierte Arbeitszuweisungen sowie automatisierte Mitteilungen

Diese neue Form der Arbeit erfordert nicht nur Softwareunterstützung, sondern auch organisatorischen Anpassungen der Abläufe und damit Veränderung des Mindsets bzw. entsprechende Schulung der Beteiligten und einheitliche Standards über die Kantons Grenzen hinweg – was angesichts der föderalen Strukturen besonders anspruchsvoll ist.

#### Professionelle Dritte: Anwälte, Gutachter und Verfahrensbeteiligte im digitalen Ökosystem

Ein oft unterschätzter, aber zentraler Bestandteil der eAktenstrategie ist die Einbindung professioneller Dritter, insbesondere:

- Rechtsanwälte: Als Hauptnutzer des elektronischen Rechtsverkehrs (ERV) müssen sie Eingaben medienbruchfrei einreichen, auf Akten zugreifen und Zustellungen empfangen können. Die neue Softwarelandschaft muss einen rollenbasierten, sicheren und barrierearmen Zugang für die Anwaltschaft bieten – auch über mobile oder webbasierte Clients.
- Gutachter, Übersetzer, Opferhilfestellen und weitere Verfahrensbeteiligte: Diese Gruppen benötigen oftmals temporären, gezielten Zugriff auf Teilbereiche der Akte. Eine moderne Lösung muss solche feingranularen Zugriffsrechte technisch sauber abbilden.
- Unternehmen: Diese Gruppe kann grosse Mengen an Strafanzeigen des gleichen Typs einreichen (zum Beispiel ÖV-Unternehmen für wiederholtes Reisen ohne gültigen Fahrausweis, Detailhandel für Diebstähle).

Die digitale Interaktion mit professionellen Dritten ist entscheidend für die Akzeptanz des neuen, durchgängig digitalen Systems. Ohne deren nahtlose Integration drohen Rückschritte und ineffiziente Mischformen aus Papier und Digital.

#### Unterschiedliche digitale Maturität der Akteure – Herausforderung für Architektur, Einführung und Betrieb

Eine grosse Herausforderung bei der Einführung der eAkte ist der unterschiedliche Digitalisierungsgrad der beteiligten Organisationen. Während einzelne Bundes- oder kantonale Behörden sowie Gerichte bereits Pilotprojekte durchführen oder moderne Fachanwendungen im Einsatz haben, arbeiten andere Behörden noch weitgehend papierbasiert. Diese heterogene Ausgangslage betrifft nicht nur die technische Infrastruktur und die damit verbundenen Prozesse, sondern auch das organisatorische Verständnis und die digitale Kompetenz der Mitarbeitenden.

Für die neue Softwarelösung bedeutet das:

- Hohe Skalierbarkeit und Anpassungsfähigkeit an spezifische funktionale und organisatorische Gegebenheiten der betroffenen Organisationseinheiten
- Flexible Einführungsmodelle: zentrale Cloudlösung vs. lokale Installation, modulweise Einführung sowie spätere Austauschmöglichkeit von Modulen
- Intuitive Benutzeroberflächen zur Reduktion von Schulungs- und Supportaufwand sowie gesteigerter Nutzung der Systeme durch den Anwender
- Kompatibilität mit bestehenden Systemen für einen schrittweisen Übergang

Diese Anforderungen zeigen: Der Paradigmenwechsel zur elektronischen Aktenführung kann nur gelingen, wenn er technologisch, organisatorisch und kulturell ganzheitlich gedacht wird. Die Einführung darf nicht nur ein IT-Projekt sein, sondern muss als gesamtjustizielle Reform verstanden und umgesetzt werden – rechtlich auf Bundesebene geregelt durch das BEKJ, das ab Inkraftsetzung die verbindliche gesetzliche Grundlage für diese Transformation bildet und den relevanten rechtlichen kantonalen Vorgaben für die Verwaltungsverfahren, wie auch den Straf- und Massnahmenvollzug.

#### 1.1.2 Grosser Anpassungsbedarf im Hinblick auf ERV

Der Elektronische Rechtsverkehr (ERV) ist ein wesentlicher Faktor für eine durchgängig digitale Justiz. Er betrifft nicht nur den formellen Austausch zwischen den Behörden, sondern auch die strukturierte Kommunikation mit externen Akteuren wie Anwälten, Behörden, Unternehmen und weiteren Verfahrensbeteiligten. Mit der Einführung des BEKJ wird der ERV zur Pflicht: Ab Inkraftsetzung müssen sämtliche Eingaben, Mitteilungen und Zustellungen digital erfolgen. Das Primat der elektronischen Aktenführung wird Realität. Der Anpassungsbedarf ist entsprechend gross:

- Standardisierung der Schnittstellen: Die bestehenden IT-Systeme bei kantonalen Behörden sowie Gerichten und bei der Anwaltschaft sind sehr heterogen. Es braucht einheitliche Schnittstellenstandards (zum Beispiel verbindliche eCH-Standards), damit Daten ausgetauscht und automatisiert verarbeitet werden können.
- Strukturierte Datenformate: Der ERV darf nicht auf PDF-Dokumente reduziert werden. Vielmehr müssen strukturierte Formate unterstützt werden (XML, JSON), um eine automatische Weiterverarbeitung – etwa für Fristenmanagement oder Entscheidungsvorschläge – zu ermöglichen.
- Zustellungsmechanismen mit Rechtsgültigkeit: Digitale Zustellung muss beweissicher, fristwährend und dokumentiert erfolgen. Das bedeutet: klare Zustellprotokolle, Zustellfiktion und Lesebestätigungen müssen abbildbar sein.
- Integration in bestehende Fachanwendungen: Der ERV muss nahtlos in die Fachlösungen der Justizbehörden integriert werden. Nur wenn Eingänge automatisch in die Aktenstruktur überführt und bearbeitet werden können, entfaltet der ERV sein volles Effizienzpotenzial.
- Sicherer Zugang für externe Nutzer: Authentifizierung über E-ID oder kantonale Zugangslösungen muss über eine zentrale Identitäts- und Rechteverwaltung gesteuert werden, um Missbrauch zu verhindern und gleichzeitig Benutzerfreundlichkeit zu gewährleisten.

Die Herausforderung liegt auch hier in der ungleichen Maturität der beteiligten Institutionen. Während einige Kantone bereits produktive ERV-Lösungen nutzen, stehen andere noch ganz am Anfang. Der Anpassungsbedarf reicht daher von der technischen Anbindung bis zur organisatorischen Neuaufstellung interner Prozesse.

Eine erfolgreiche Fachapplikation muss daher modular, interoperabel und erweiterbar gestaltet sein – mit zentralen Plattformdiensten (wie einheitlicher Zustellservice, Fristenkalender, Aktenregistratur) und dezentralen Fachkomponenten. Nur so kann der ERV seinen Beitrag zur Effizienz, Transparenz und Verfahrenssicherheit in der Schweizer Justiz leisten.

### 1.1.3 Generell sind die bestehenden Lösungen veraltet und müssen erneuert werden.

Ein kritischer Erfolgsfaktor ist der Zustand der bestehenden IT-Systemlandschaften: Viele der heute verwendeten Fachanwendungen sind technologisch überaltert, schwer wartbar und nicht auf künftige Anforderungen vorbereitet. Dieser Befund zieht sich durch nahezu alle Ebenen der kantonalen und eidgenössischen Justizinformatik.

Zahlreiche Systeme stammen aus einer Zeit, in der papierbasierte Verfahren noch konkurrenzlos waren. Entsprechend ist ihre Architektur oft nicht auf digitale Akten- und Verfahrensführung ausgelegt. Beispiele hierfür sind monolithische Anwendungen mit proprietären Datenstrukturen, eingeschränkter Modularität und fehlender oder eingeschränkter Mandantenfähigkeit.

Weitere typische Merkmale der in die Jahre gekommenen Systemlandschaft:

- Fehlende Integrationsfähigkeit mit modernen Plattformdiensten wie ERV, Identitätsmanagement oder Cloud-Infrastrukturen
- Mangelnde Benutzerfreundlichkeit: veraltete Benutzeroberflächen, komplizierte Bedienung und dementsprechend hohem Schulungsaufwand
- Grosser Pflegeaufwand: aufwändige individuelle Anpassungen, fehlende Automatisierung von Test- und Releaseprozessen, unzureichende Dokumentation
- Begrenzte Weiterentwicklungsmöglichkeiten: aufgrund fehlender APIs, proprietärer Technologien oder mangelnder Ressourcen der Anbieter

Diese Ausgangslage führt dazu, dass viele Behörden an einem „digitalen Limit“ operieren – mit hohem manuellem Aufwand, zunehmenden Sicherheitsrisiken und steigenden Betriebskosten. Gleichzeitig können innovative Funktionen wie Analytics, intelligente Assistenzsysteme oder automatisierte Entscheidungsunterstützung gar nicht oder nur mit erheblichem Aufwand integriert werden.

Ein technischer Neuanfang ist deshalb unabdingbar. Dieser muss nicht nur die bestehende Funktionalität abdecken, sondern ein zukunftsgerichtetes, erweiterbares und standardbasiertes Fundament für die digitale Justiz schaffen. Nur so kann der notwendige Kulturwandel in der Verwaltung durch eine entsprechende technologische Basis gestützt werden.

### 1.1.4 Sehr oft monolithischer Aufbau

Ein zentrales architektonisches Problem der bestehenden Softwarelösungen in der Justiz liegt im monolithischen Aufbau vieler Fachanwendungen. Dabei handelt es sich um Systeme, die als grosse, geschlossene Einheiten entwickelt wurden, in denen sämtliche Funktionalitäten – von der Benutzeroberfläche über die Geschäftslogik bis zur Datenhaltung – eng miteinander verknüpft sind und als sogenannt monolithisches Gesamtsystem funktionieren.

#### Nachteile monolithischer Strukturen

Der monolithische Aufbau bringt aus heutiger Sicht erhebliche Nachteile mit sich:

- Eingeschränkte Skalierbarkeit: Die Systeme lassen sich schlecht auf moderne Infrastrukturen (zum Beispiel Cloud, Containerisierung) übertragen, da einzelne Module nicht unabhängig betrieben werden können.
- Fehlende Flexibilität bei Erweiterungen: Jede Erweiterung oder Anpassung am System erfordert Eingriffe in die zentrale Codebasis. Das erhöht den Entwicklungsaufwand und das Risiko unbeabsichtigter Nebeneffekte.
- Verzögerte Releases: Aufgrund komplexer Abhängigkeiten können Änderungen nur in grossen, risikobehafteten Releasezyklen ausgerollt werden. Agile oder iterative Entwicklungsansätze sind kaum realisierbar.
- Technologische Abhängigkeit: Viele Monolithen basieren auf proprietären Technologien, die schwer oder gar nicht mehr wartbar sind. Eine Migration auf moderne Architekturen ist oft nur durch komplette Neuentwicklung möglich.

- Hindernis für föderalen Betrieb: Ein monolithisches System lässt sich nur schwer modular auf die unterschiedlichen Bedürfnisse von verschiedenen Kantonen oder Behörden zuschneiden – was im föderalen Kontext der Schweiz ein gravierender Nachteil ist.

#### Hindernis für Integration und Innovation

Monolithische Anwendungen erschweren zudem die Integration externer Systeme, wie beispielsweise ERV-Plattformen, Identitätsdienste, Dokumentenmanagementsysteme oder Analysewerkzeuge. Moderne Anforderungen wie Mehrmandantenfähigkeit, rollenbasierter Zugriff oder Anbindung mobiler Endgeräte sind in monolithischen Architekturen meist nur unzureichend realisierbar.

Auch Innovationen – etwa durch Einsatz von KI-gestützten Recherchetools, digitalen Assistenten oder Prognosemodellen – lassen sich kaum einbetten, da der monolithische Kern keine flexiblen Integrationspunkte bietet. Damit wirken monolithische Systeme wie eine digitale „Einbahnstrasse“, die fortlaufend Wartung erfordert, aber kaum Weiterentwicklung ermöglicht.

#### Notwendigkeit eines modularen Architekturparadigmas

Die Ablösung monolithischer Systeme zugunsten eines modularen, serviceorientierten Architekturmodells (zum Beispiel Microservices oder modulare Domänenarchitekturen) ist deshalb eine der zentralen Voraussetzungen für eine zukunftsfähige Justizlösung. Dabei sollten folgende Prinzipien leitend sein:

- Entkopplung der Komponenten: Jede Funktionseinheit (zum Beispiel Aktenführung, Fristenmanagement, Zustellung) wird als eigenständiger Dienst umgesetzt.
- Klare APIs und Schnittstellenstandards: Alle Komponenten kommunizieren über standardisierte Schnittstellen, welche die Integration externer Systeme ermöglichen.
- Modularer Rollout: Neue Funktionen oder Updates können schrittweise eingeführt werden, ohne das Gesamtsystem zu beeinflussen.
- Skalierbarkeit und Ausfallsicherheit: Moderne Betriebsmodelle auf Basis von Cloud-Technologien können flexibel an Lasten angepasst und hochverfügbar betrieben werden.

Dieser Paradigmenwechsel auf Architekturebene ist essenziell, um Technologie als Enabler der Justizreform zu positionieren und nicht als limitierenden Faktor wahrzunehmen. Nur durch eine konsequent modulare Struktur kann die Software den organisatorischen Anforderungen des Justizwesens und dessen Veränderungen langfristig gerecht werden.

#### 1.1.5 Wenig flexible Architektur

Neben dem monolithischen Aufbau weisen viele der heutigen Justizlösungen eine technisch und organisatorisch unflexible Architektur auf. Diese mangelnde Flexibilität ist einer der Hauptgründe dafür, dass bestehende Systeme weder mit neuen Anforderungen wachsen können noch effizient auf gesetzliche oder organisatorische Veränderungen reagieren.

#### Fehlende Modularität und Erweiterbarkeit

Ein wesentliches Problem liegt in der engen Kopplung von Geschäftslogik, Benutzeroberfläche und Datenhaltung. Funktionen sind häufig hart codiert, Erweiterungen benötigen tiefgreifende Eingriffe in die Systembasis. Dies verhindert:

- die einfache Integration neuer Module, zum Beispiel für elektronische Beweisführung, Videoverhandlungen oder automatisiertes Fristenmanagement
- den parallelen Betrieb unterschiedlicher Prozesslogiken, zum Beispiel für verschiedene Verfahrenstypen oder kantonale Besonderheiten
- den einfachen Austausch einzelner Komponenten, etwa von Benutzeroberflächen oder Datenbanken, ohne das Gesamtsystem zu beeinträchtigen

Stattdessen müssen bereits kleine Erweiterungen oder Anpassungen in langwierigen Projekten entwickelt und getestet werden – oft verbunden mit erheblichen Kosten und Verzögerungen.



### Reaktionsunfähigkeit auf regulatorische Änderungen

Gerade im Bereich der Justiz ist die IT eng mit dem Rechtssystem und dessen laufender Entwicklung verknüpft. Neue gesetzliche Vorgaben – wie sie durch das BEKJ, durch Entscheidungen der Gesetzgeber oder Weiterentwicklung der bestehenden Prozessordnungen entstehen – erfordern Anpassungen der Systeme.

Unflexible Architekturen stehen dieser Dynamik entgegen. Änderungen müssen aufwändig in bestehende Logikblöcke eingebaut werden, was zu erhöhtem Testing- und Validierungsaufwand führt. Dies hemmt die Anpassungsfähigkeit und gefährdet die Rechtssicherheit, wenn Fristen oder gesetzliche Anforderungen nicht rechtzeitig technisch umgesetzt werden können.

### Limitierung durch fehlende Konfigurierbarkeit

Manche Systeme lassen kaum Raum für konfigurierbare Fachlogik oder parametrisierbare Abläufe. Damit sind die betroffenen Organisationen selbst für einfache Änderungen – zum Beispiel neue Vorlagen, Rollenrechte, Protokollstandards oder Notifikationsregeln – auf die Hersteller oder Systemintegratoren angewiesen.

Dies führt zu einer ausgeprägten technologischen Abhängigkeit und verhindert, dass Fachbereiche selbständig auf sich ändernde Bedürfnisse reagieren können. Ein modernes System hingegen sollte über ein umfassendes Regel- und Konfigurationswerk verfügen, das von befähigten Justizmitarbeitenden oder der zuständigen (eigenen) Supportorganisation ohne Programmierung angepasst werden kann.

### Anforderungen an eine moderne, flexible Architektur

Eine zukunftsgerichtete Justizlösung muss daher folgende Prinzipien erfüllen:

- Hohe Parametrisierbarkeit: Fachlogik, Workflows, Rollen und Zuständigkeiten müssen durch Konfiguration steuerbar sein.
- Multi-Tenant- und Mandantenfähigkeit: Staatsanwaltschaften, Gerichte und Akteure des Straf- und Massnahmenvollzugs sollen dieselbe Lösung nutzen können, ohne sich gegenseitig zu beeinflussen.
- Einfache Anbindung externer Systeme: über offene Schnittstellen, moderne Protokolle und eventbasierte Kommunikation
- Modularer Aufbau mit klarer Trennung von Zuständigkeiten: zum Beispiel Trennung von Core-Funktionalität, Fachlogik, Frontend und Integrationsdiensten

Nur mit einer solchen Architektur kann die Softwarelösung zur tragfähigen Plattform für eine über Jahre andauernde, dynamische Digitalisierung der Justiz werden – anpassbar, offen, nachhaltig und nutzerzentriert. Die damit verbundenen betrieblichen Anforderungen bzw. das Zusammenspiel der Softwarearchitektur mit den betrieblichen Rahmenbedingungen werden im Kapitel Betriebsmodelle aufgezeigt.

#### 1.1.6 Anspruchsvoller Unterhalt und eingeschränktes Potential zur Weiterentwicklung

Ein zentrales Problemfeld der aktuellen Justizinformatik ist nicht nur die veraltete Softwarearchitektur, sondern auch die damit verbundene Abhängigkeit von spezialisierten Personen bei den bestehenden Lieferanten und den Lieferanten selbst. Über Jahre hinweg wurden Fachanwendungen durch IT-Dienstleister entwickelt und gepflegt, deren Mitarbeitende über ein einzigartiges Zusammenspiel aus technischem Know-how und tiefem justizspezifischem Fachwissen verfügten. Diese Experten kannten nicht nur den Code, sondern auch die juristischen Prozesse, Rollen und Besonderheiten der verschiedenen Organisationen im Detail. Mit dem zunehmenden Generationenwechsel, dem Rückzug etablierter Anbieter und dem strukturellen Wandel im IT-Dienstleistungsmarkt ist dieses personengebundene Wissen jedoch zunehmend gefährdet. Viele dieser zentralen Fachpersonen stehen heute nicht mehr zur Verfügung, was die Wartung, Weiterentwicklung und den Know-how-Transfer im Kontext bestehender Lösungen erheblich erschwert – teilweise bis hin zur faktischen Unveränderbarkeit der Lösung. Gleichzeitig geraten die Behörden in eine riskante Abhängigkeit von schwer ersetzbarem Individualwissen ausserhalb ihrer eigenen Organisation.

Weiter drängen neue Anbieter und technologische Quereinsteiger auf den Markt – etwa Start-ups, etablierte IT-Dienstleister und Lieferanten. Diese verfügen zwar über moderne Entwicklungsansätze und erprobte Technologien, bringen aber oft wenig bis kein spezifisches Fachverständnis für die Schweizer Justiz mit. Das Fehlen von juristischem und prozeduralem Domänenwissen ist aus Sicht der neuen Anbieter kein Hindernis, um IT-Dienstleistungen zu erbringen – für die Justizbehörden jedoch führt das zu einem erhöhten Koordinations- und Steuerungsaufwand.

Diese Entwicklung hat unmittelbare Konsequenzen für die Betreiberrolle:

- Die Justizbehörden oder deren IT-Dienstleister müssten mehr Fachwissen selbst aufbauen, um die neuen Lieferanten steuern, Anforderungen präzise formulieren und Qualität sichern zu können.
- Sie müssen in der Lage sein, Architekturentscheide zu treffen, Schnittstellen und Datenmodelle zu bewerten und selbstständig Test- und Abnahmeprozesse durchzuführen.
- Sie benötigen zunehmend betriebliches Know-how über die eingesetzten Lösungen, um bei Störungen, Erweiterungswünschen oder Anpassungen nicht vollständig vom Lieferanten abhängig zu sein.

Dies wirft eine grundlegende strategische Frage auf: Soll das Beherrschen und Weiterentwickeln von Fachapplikationen zur Kernkompetenz einer Justizbehörde oder IT-Organisation gehören? Oder soll diese Aufgaben durch einen behördlichen oder industriellen Generalunternehmer gelöst werden? Besteht dadurch eine "leichter" zu akzeptierende Abhängigkeit? Die Antwort darauf ist nicht einheitlich und muss je nach Kontext, Ressourcen und digitaler Strategie individuell beantwortet werden.

Fest steht jedoch: Je höher die fachliche Unabhängigkeit vom Lieferanten, desto grösser sind die Anforderungen an interne Organisationen – sowohl personell als auch strukturell. Dies betrifft unter anderem:

- Schulung und langfristige Bindung von juristischem Fachpersonal mit IT-spezifischem Wissen
- Schulung und langfristige Bindung von IT-Fachpersonal mit justizspezifischem Wissen
- Aufbau von systematischem Lösungs- und Betriebswissen
- Einführung von standardisierten Betriebshandbüchern, Architektur- und Governance-Vorgaben
- Klare Definition der Verantwortlichkeiten zwischen Betreiber, Dienstleister und Behörde

In der Praxis zeigt sich zudem, dass veraltete, schwer verständliche Systeme oft nicht mehr wirtschaftlich modernisiert werden können, weil kein Know-how mehr vorhanden ist oder die Technologie nicht mehr unterstützt wird. Gleichzeitig verhindern fragmentierte Zuständigkeiten die schnelle Umsetzung innovativer Funktionen, etwa zur Automatisierung, Prozessanalyse oder digitalen Zusammenarbeit.

Um dieser Entwicklung zu begegnen, braucht es ein zukunftsfähiges Betriebs- und Entwicklungsmodell, das auf modularen Lösungen mit klar dokumentierter Schnittstellen- und Datenarchitektur basiert. Nur so können Behörden unabhängig von einzelnen Personen oder Anbietern agieren – und die Justizinformatik in eine nachhaltige, wartbare und evolvierbare Zukunft führen.

## 1.2 Nutzerorganisationen / Anwenderseite

Die Nutzerseite zeichnet sich derzeit durch folgende Merkmale aus:

Die Nutzerorganisationen sind gemäss föderaler Struktur sehr stark segmentiert und verfügen dadurch nur über eine schwache Nachfragemacht. Diese Segmentierung wirkt sich bei der Beschaffung nachteilig aus. Mit der Einführung des ERV steigt die Wichtigkeit einer modernen Geschäftsverwaltung. Damit verbunden wächst in den meisten Organisationen das Bewusstsein für die strategische Bedeutung von modernen IT-Lösungen. Um diese Herausforderungen gerecht zu werden, müssen entsprechend kompetente Spezialisten auf Behördenseite vorhanden sein, um diesem Anspruch nicht nur im Betrieb sondern auch in der strategischen Steuerung gerecht zu werden. Viele Behörden sind aber oft auf die Bewältigung des Tagesgeschäfts ausgerichtet und nicht auf den kontinuierlichen Change-Prozesse über die nächsten 20-30 Jahre. Trotzdem besteht grosser Anpassungs- und Erneuerungsdruck mit engen Zeitvorgaben und knappen Budgets. Die teilweise fehlende Kompetenz im technischen Bereich führt zu grosser Abhängigkeit von Lieferanten und externen Beratern. Zudem stellt der Betrieb von komplexen Applikationen hohe Anforderungen an die Anwenderorganisationen, die aufgrund fehlender eigener Ressourcen nur ungenügend abgedeckt werden können. Durch grössere Fähigkeitslücken kann zudem die zeitgerechte und ordnungsgemässe Abwicklung der Rechtsgeschäfte erschwert und dadurch das Vertrauen in den Rechtsstaat beeinträchtigt werden.

## 1.3 Lieferantensituation

Die Schweizer Justizinformatik bewegt sich in einem äusserst fragmentierten Marktumfeld, das für Softwarelieferanten und Dienstleister in mehrfacher Hinsicht anspruchsvoll ist. Die föderale Struktur, kantonale Unterschiede in der Verfahrensausgestaltung, sprachliche Vielfalt sowie die divergierenden Digitalisierungsvoraussetzungen führen dazu, dass der Markt trotz seines relativ kleinen Gesamtvolumens hohe Komplexität und heterogene Anforderungen mit sich bringt.

Für Anbieter bedeutet dies:

- Jede Ausschreibung, jede Einführung und jede Weiterentwicklung ist hochspezifisch und mit erheblichem Anpassungsaufwand verbunden.
- Das Standardisierungspotenzial ist bisher ungenügend ausgeschöpft, was eine wirtschaftliche Skalierung von Lösungen erschwert.
- Die Anzahl potenzieller Kunden pro Produktvariante ist beschränkt – insbesondere, wenn sie auf einen spezifischen Kanton oder Fachbereich zugeschnitten ist.

Daraus ergibt sich ein zentrales Dilemma: Mittlere und kleinere Nutzerorganisationen – also viele kantonale Ämter, kleinere Gerichte oder spezialisierte Fachstellen – sind für Anbieter kommerziell oft nur begrenzt attraktiv. Der Aufwand für Einführung, Support und Anpassung steht häufig in keinem Verhältnis zum Umsatzpotenzial. Diese Situation führt zu zwei problematischen Tendenzen:

- Verzögerte oder ganz ausbleibende Innovation: Anbieter sind wenig motiviert, für kleine Nutzergruppen spezifische Funktionen oder Integrationen zu entwickeln. Neue Technologien oder Usability-Verbesserungen erreichen diese Organisationen nur verzögert oder gar nicht.
- Zunehmende Abhängigkeit von wenigen etablierten Anbietern: Es entsteht eine Tendenz zur Konzentration im Anbieterfeld, insbesondere im Justiz- und Sicherheitsbereich. Wenige Unternehmen dominieren den Markt und konsolidieren ihre Position durch langfristige Wartungsverträge, Know-how-Vorsprung oder die Übernahme von kleineren Wettbewerbern.

Diese Monopol- oder Oligopol-situation birgt Risiken:

- Geringere Innovationsdynamik und steigende Preise
- Abhängigkeit von spezifischem Fachpersonal innerhalb des Lieferantenunternehmens
- Weniger Auswahlmöglichkeiten bei Neuausschreibungen
- Hohes Risiko bei Lieferantenwechsel oder Personalfluktuations auf Seiten des Anbieters

Gleichzeitig treten neue Player in den Markt, die zwar über moderne Technologien, aber wenig domänen-spezifisches Wissen im Bereich Justiz verfügen. Diese Anbieter können wertvolle Impulse bringen – etwa im Bereich Cloud-Architektur, Low-Code-Plattformen oder KI-gestützter Auswertung –, benötigen aber auf Behördenseite ein deutlich höheres Mass an Steuerung, Fachkoordination und Qualitätssicherung.

Diese Dynamik führt dazu, dass Betriebsorganisationen – etwa kantonale Fachämter, Informatikdienste oder übergeordnete Stellen – vermehrt selbst tiefes Fach- und Betriebswissen aufbauen müssen, um sowohl etablierte wie neue Anbieter kompetent führen zu können.

Die zentrale Frage lautet dabei: Ist dies (noch) eine Kernkompetenz öffentlicher Organisationen – oder braucht es neue Kooperations- oder Zentralisierungsmodelle auf nationaler Ebene?

Langfristig lässt sich festhalten:

- Der Schweizer Markt muss sich aus Anbieter- wie Nutzersicht standardisieren und vereinheitlichen, um Innovationen wirtschaftlich tragfähig zu machen.
- Nutzerorganisationen müssen in der Lage sein, Lieferanten strategisch zu steuern und technologische Wechsel aktiv mitzugestalten.
- Eine professionelle Governance-Struktur und Plattformstrategie wird zur notwendigen Voraussetzung, um aus der heutigen Anbieterabhängigkeit auszubrechen und digitale Souveränität zurückzugewinnen.

#### 1.4 Fazit und Folgerungen

Die Analyse der aktuellen Situation in der Schweizer Justizinformatik zeigt einen klaren Handlungsbedarf: Die bisher eingesetzten, oft monolithischen und technologisch veralteten Fachanwendungen sind nicht mehr geeignet, die umfassenden Anforderungen einer durchgängig digitalen und effizienten Justiz abzubilden. Der Paradigmenwechsel hin zur elektronischen Aktenführung, die zunehmende Bedeutung des elektronischen Rechtsverkehrs sowie die gesetzlichen Vorgaben des BEKJ ab dessen Inkraftsetzung erfordern eine grundlegende Erneuerung der IT-Landschaft.

#### 1.4.1 Notwendigkeit neuer Anbieter mit modernem Ansatz

Der Markt für Justizinformatik befindet sich im Umbruch: Klassische Lieferanten, die über umfangreiches domänenspezifisches Wissen und technisches Know-how verfügen, sind nur noch in begrenztem Umfang verfügbar. Neue Anbieter drängen auf den Markt, deren Stärke weniger im Justizfachwissen, sondern in innovativen Technologien und modernen Entwicklungs- und Betriebsansätzen liegt. Dies eröffnet Chancen für frische Impulse, stellt aber auch Anforderungen an den Wissenstransfer und die Zusammenarbeit, damit fachliche Anforderungen zielgerichtet umgesetzt werden können.

#### 1.4.2 Wissenstransfer und Domänenkompetenz als Schlüsselfaktor

Der Erfolg einer zukunftsgerichteten Lösung hängt entscheidend davon ab, dass das spezifische Fachwissen der Justiz an die neuen Anbieter übergeben und in die Systeme integriert wird. Gleichzeitig müssen die Betreiberorganisationen – Behörden und IT-Betriebe – ihre Kompetenzen im Bereich moderner Technologien ausbauen, um den Betrieb zukunftssicher und unabhängig gestalten zu können. Diese Balance zwischen fachlicher Expertise und technischer Kompetenz ist essenziell.

#### 1.4.3 Risiko der Lieferantenabhängigkeit minimieren

Die bisherige Praxis, bei der Lieferanten umfassendes Spezialwissen bündelten und damit eine hohe Abhängigkeit entstanden ist, muss überwunden werden. Eine moderne Architektur, die auf modularen und offenen Standards basiert, reduziert die Bindung an einzelne Anbieter und ermöglicht flexiblere Beschaffungs- und Weiterentwicklungsstrategien.

#### 1.4.4 Best-of-Breed-Ansatz als erfolgversprechende Strategie

Eine technische Zielarchitektur, die nach dem Best-of-Breed-Prinzip spezialisierte Komponenten kombiniert, bietet die notwendige Flexibilität, um die unterschiedlichen Anforderungen der heterogenen Schweizer Justizlandschaft zu erfüllen. Durch die Kombination bewährter Einzelanwendungen lassen sich sowohl Investitionsschutz als auch Innovationsfähigkeit gewährleisten.

## 2 Zielsetzung Studie

Die vorliegende Konzeptstudie verfolgt das Ziel, eine fundierte Entscheidungsgrundlage für die zukünftige Ausgestaltung von Fachapplikationen im Schweizer Justizbereich zu schaffen. Im Mittelpunkt steht die Entwicklung eines tragfähigen, langfristigen Lösungsansatzes, der den aktuellen und zukünftigen Anforderungen an Effizienz, Flexibilität, Sicherheit und Nachhaltigkeit gerecht wird. Die Zielsetzung gliedert sich in drei zentrale Schwerpunkte: Technische Ausarbeitung einer langfristigen Lösung, Machbarkeitsanalyse einer Best-of-Breed-Strategie und Chancen- und Risikoanalyse der vorgeschlagenen Lösung.

### 2.1 Technische Definition einer langfristigen Lösung

Ein zentrales Anliegen dieser Studie ist die Entwicklung einer technischen Zielarchitektur, die als Alternative zu den bestehenden, oftmals monolithischen und nicht mehr zeitgemässen Fachapplikationen dient. Die neue Lösung soll modular, skalierbar und offen für zukünftige technologische Entwicklungen sein. Sie muss den Paradigmenwechsel von papierbasierten zu vollständig digitalen Prozessen entlang der gesamten Justizkette unterstützen und dabei sowohl die föderalen Strukturen als auch die unterschiedlichen Digitalisierungsgrade der beteiligten Organisationen berücksichtigen.

Die technische Zieldefinition umfasst insbesondere:

- Die Beschreibung einer modularen Architektur, die lose gekoppelte Komponenten und klare Schnittstellen vorsieht
- Die Berücksichtigung von Anforderungen an Sicherheit, Datenschutz, Mandantenfähigkeit und Revisions-sicherheit
- Die Definition von möglichen organisatorischen wie auch technischen Betriebsmodellen der Lösung

## 2.2 Machbarkeitsanalyse einer Best-of-Breed-Strategie

Ein weiterer Schwerpunkt der Studie ist die Bewertung, inwiefern eine Best-of-Breed-Strategie, also die Kombination spezialisierter Anwendungen und Komponenten, eine geeignete und nachhaltige Lösung für den Justizbereich darstellt. Ziel ist es, die Vorteile spezialisierter, am Markt bewährter Systeme zu nutzen und diese über standardisierte Schnittstellen zu einer leistungsfähigen Gesamtlösung zu integrieren. Die Bewertung der Best-of-Breed-Strategie erfolgt auf Basis folgender Aspekte:

- Die Untersuchung, wie verschiedene spezialisierte Komponenten (zum Beispiel Dokumentenmanagement, Fristenmanagement, Statistik oder Kommunikation) technisch und organisatorisch zu einer Gesamtlösung verbunden werden können. Dabei werden die Anforderungen an Schnittstellen, Datenflüsse und die Integration in bestehende Systemlandschaften betrachtet
- Die Identifikation von Erfolgsfaktoren und potenziellen Stolpersteinen bei der Umsetzung einer Best-of-Breed-Architektur im Justizumfeld
- Die Ableitung von Empfehlungen für die Auswahl, Integration und Weiterentwicklung von Komponenten. Die Empfehlungen basieren auf den Erkenntnissen aus der Analyse der bestehenden Systemlandschaft, den Rückmeldungen aus Workshops und Interviews sowie auf Erfahrungen aus vergleichbaren Projekten.

## 2.3 Chancen- und Risikoanalyse

Zu guter Letzt soll die Studie Chancen- und Risiko analysieren, um die Tragfähigkeit und Zukunftssicherheit der vorgeschlagenen Lösung zu bewerten. Ziel ist es, sowohl die Potenziale als auch die Herausforderungen einer modularen, ereignisorientierten und Best-of-Breed-basierten Architektur transparent zu machen.

Im Rahmen der Chancen- und Risikoanalyse werden folgende Aspekte betrachtet:

- Die Identifikation von Chancen, insbesondere im Hinblick auf Flexibilität, Innovationspotenzial und die Möglichkeit zur schrittweisen Modernisierung
- Die Bewertung des Investitionsschutzes durch die Nutzung offener Standards und die Möglichkeit, einzelne Komponenten unabhängig voneinander zu erneuern oder auszutauschen
- Die Analyse von Risiken, etwa im Hinblick auf Integrationsaufwand, technische Komplexität, Governance, Betriebssicherheit und die nachhaltige Pflege von Open-Source-Komponenten
- Die Formulierung von Empfehlungen, wie Risiken adressiert werden können, zum Beispiel durch klare Schnittstellen, verbindliche Standards, Know-how-Aufbau und die Etablierung von Governance-Strukturen

## 2.4 Methodik

Die Zielsetzungen der Konzeptstudie wurden durch ein mehrstufiges methodisches Vorgehen adressiert. Zunächst erfolgte eine umfassende Analyse der bestehenden Systemlandschaft und der aktuellen Herausforderungen im Schweizer Justizbereich. Darauf aufbauend wurden Anforderungen und Erwartungen der Nutzer und Organisationen erhoben, insbesondere durch Workshops, Interviews und die Auswertung von Rückmeldungen aus dem Markt sowie von Vertretern verschiedener Justizbehörden und potenziellen Anbietern.

Die technische Zielarchitektur wurde auf Basis bewährter Prinzipien aus der Softwareentwicklung und unter Berücksichtigung der spezifischen Anforderungen des Justizumfelds entwickelt und im Rahmen der Studie diskutiert. Die Chancen- und Risikoabschätzung erfolgte auf Grundlage der gesammelten Erkenntnisse, der Rückmeldungen aus den Interviews sowie durch den Vergleich mit Erfahrungen aus ähnlichen Digitalisierungsprojekten im öffentlichen Sektor.

Dieses methodische Vorgehen bildet die Grundlage für die im weiteren Verlauf der Studie dargestellten Empfehlungen und Entscheidungsgrundlagen.

## 3 Situation bei den Fachapplikationen und Lieferanten

Um die in Kapitel 2 definierte Zielsetzung zu erreichen, ist ein fundiertes Verständnis der aktuellen Systemlandschaft erforderlich. Dieses Kapitel beschreibt die bestehende Situation im Hinblick auf die eingesetzten Fachapplikationen sowie die beteiligten Lieferanten. Die Analyse dient als Grundlage, um





Einsatz. Dies zeigt, dass es innerhalb eines Kantons nicht nur Unterschiede zwischen den Justizbereichen gibt. Selbst innerhalb eines Bereichs, wie hier bei den Gerichten, werden unterschiedliche Fachapplikationen eingesetzt.

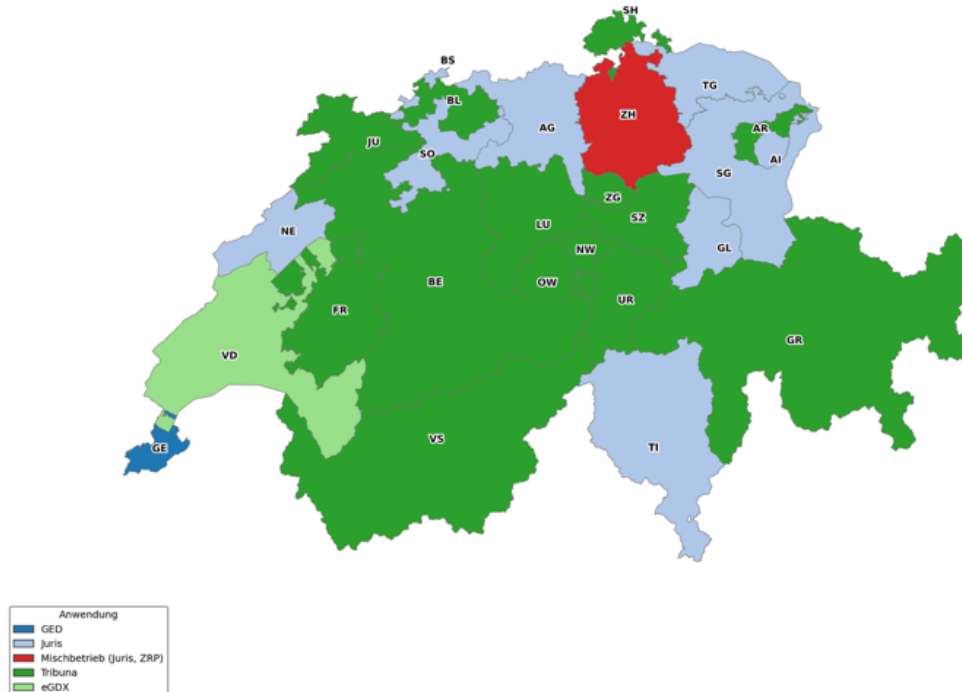


Abbildung 2 Übersicht Fachapplikationen Gerichte

### 3.1.3 Justizvollzugsbehörden

Die Abbildung 3 zeigt nun die Verteilung von Applikation in den Justizvollzugsbehörden. Dabei zeigt die Karte sowohl JUV-Ämter wie auch Institutionen. Im Vergleich zu den Gerichten und Staatsanwaltschaften ist hier Tribuna nicht vertreten. Diese bildet den Teil der Justizvollzugsbehörden in der Software nicht ab. Dafür ist eine andere Software vorhanden. GINA von der Glaux Group AG. Das Produkt ist speziell für den Justizvollzug entwickelt und bildet dort die gesamte Prozesskette ab. Es dominiert diesen Teil des Marktes und wird in 16 von 26 Kantonen eingesetzt. In 14 Kantonen setzt man dabei nur auf das Produkt GINA im Justizvollzug. Die zweitmeist verbreitete Applikation ist wie bei den Gerichten und der Stawa, das Produkt JURIS. Es wird in 7 Kantonen eingesetzt. Die Kantone Basel-Stadt, Thurgau, Tessin und Neuenburg setzen dabei vollkommen auf die Lösung der Unternehmung LogObject AG. Einzig und alleine die Kantone Genf, Waadt, Zürich und Schaffhausen kommen ohne eines der beiden Produkte aus. Genf und Waadt setzen hier zum Beispiel auf die Lösung Papillon und Zürich auf RIS. Es gibt also auch hier eine hohe Konzentration aber auch bei den Justizvollzugsbehörden fällt auf dass es einige Kantone mit Mischbetrieb von mehreren Fachapplikationen gibt.

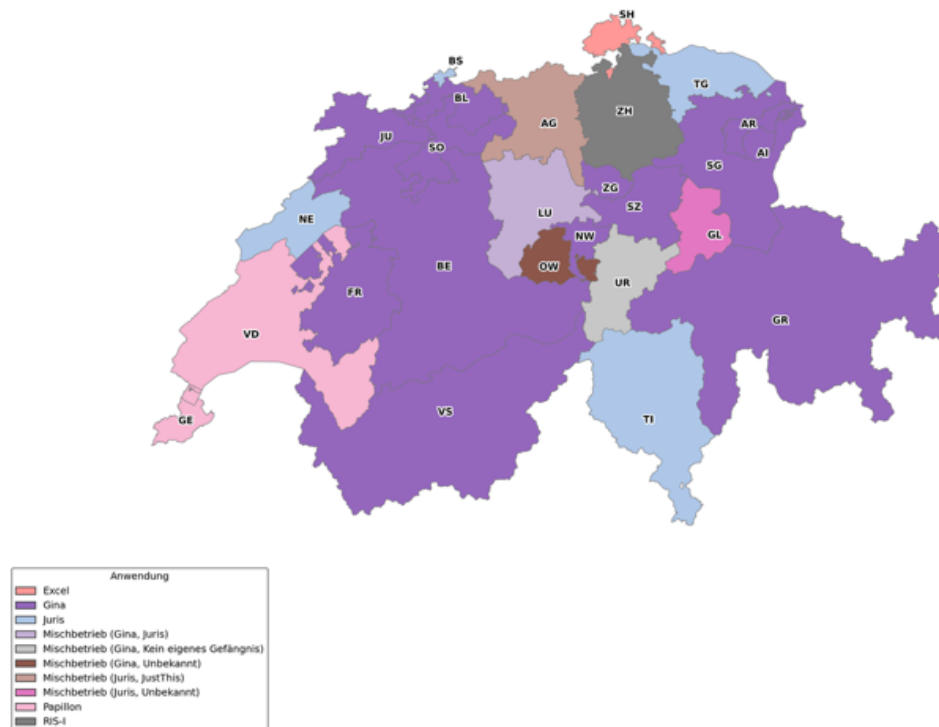


Abbildung 3 Übersicht Fachapplikationen Justizvollzugsbehörden

### 3.2 Fazit zur Situation bei den Fachapplikationen

Die Analyse zeigt eine heterogene Fachapplikationslandschaft im Justizbereich. Im Einsatz sind zu einem grossen Teil einige führende Produkte wie JURIS, Tribuna oder GINA im Justizvollzug, dennoch werden selbst innerhalb von Kantonen unterschiedliche Produkte verwendet. Diese Unterschiede treten nicht nur zwischen den Bereichen Staatsanwaltschaft, Gerichte und Justizvollzug auf, sondern teilweise sogar innerhalb derselben Bereiche. Die daraus resultierende Vielfalt führt zu parallelen Strukturen und Doppelspurigkeiten, die den Betriebsaufwand und die Gesamtkosten für die Justiz erheblich erhöhen.

Viele der genutzten Systeme sind individuell an die spezifischen Prozesse der jeweiligen Behörde angepasst. Customizing ist für eine sinnvolle Nutzung essenziell, da derzeit keine standardisierten Prozesse schweizweit existieren. Behörden desselben Typs können für die Erledigung einer Aufgabe völlig unterschiedliche Abläufe haben. So wird beispielsweise Tribuna sowohl in den Staatsanwaltschaften von Bern als auch in Fribourg eingesetzt, jedoch meist so angepasst, dass die Entwicklung neuer Funktionen vollständig auf den jeweiligen Einsatzort zugeschnitten werden muss. Dies macht Weiterentwicklungen entsprechend aufwändig. Gleichzeitig führt die hohe Anpassung zu einer extremen Passgenauigkeit: Die Software bildet die jeweiligen operativen Prozesse sehr exakt ab und ermöglicht dadurch eine hohe Effizienz in der täglichen Arbeit.

Diese Spezialisierung erfordert seitens der Anbieter tiefgehende Fachkenntnisse und umfassendes Prozesswissen, was eine wesentliche Ursache für die starke Marktkonzentration auf wenige Anbieter darstellen könnte. Diese Kenntnisse bieten Chancen, da die Anbieter stark spezialisiert sind und fehlende Kompetenzen auf Seiten der Betriebsorganisationen ausgleichen können. Gleichzeitig birgt diese Situation erhebliche Risiken. Die Abhängigkeit von wenigen Anbietern kann die Verhandlungsposition der öffentlichen Hand schwächen und die Flexibilität bei zukünftigen Anpassungen einschränken. Lock-in-Effekte entstehen, wenn ein Wechsel des Systems aufgrund hoher Migrationskosten oder fehlender Alternativen kaum realisierbar ist. Darüber hinaus kann die Innovationsgeschwindigkeit leiden, da die Anbieter aufgrund ihrer dominanten Marktstellung weniger Anreize für disruptive Weiterentwicklungen haben. Strategische Risiken ergeben sich auch im Hinblick auf die langfristige Verfügbarkeit und Preisgestaltung, die stark von den Entscheidungen einzelner Anbieter abhängen.

Für die Zukunft könnte eine koordinierte Strategie die Möglichkeit bieten, Kosten zu senken, Prozesse zu vereinfachen und die Interoperabilität zu verbessern, wobei kantonale Besonderheiten berücksichtigt werden müssen. Daraus ergeben sich zentrale Handlungsbedarfe wie zum Beispiel die Harmonisierung von Prozessen, um Doppelspurigkeiten zu vermeiden und den jeweiligen Anpassungsaufwand zu reduzieren.



Eine solche Initiative würde nicht nur die Effizienz steigern, sondern auch die Risiken der heutigen heterogenen Landschaft deutlich reduzieren. Ergänzend sollte auf Behördenseite entsprechendes Know-how aufgebaut werden, sodass Anbieter künftig nicht mehr über ein so tiefes Prozess- und Fachverständnis verfügen müssen. Dies würde den Eintritt weiterer Anbieter in den Markt erleichtern und die Abhängigkeit von den wenigen dominanten Akteuren reduzieren.

## 4 Bedürfnisse und Herausforderungen auf Anwenderseite

Neben den Herausforderungen auf dem Lieferantenmarkt stellt sich für die Justizbehörden auch die Frage der Erwartungen ihrer Mitarbeitenden. Diese arbeiten täglich mit der Fachapplikation und haben ebenfalls spezifische Erwartungen und Anforderungen an eine neue Fachapplikation. Ihre Erwartungen orientieren sich stark an den Erfahrungen aus dem privaten digitalen Umfeld, das zunehmend durch intuitive Bedienbarkeit, Automatisierung oder auch den Einsatz von künstlicher Intelligenz geprägt ist. Dabei ist es wichtig, userzentriert zu arbeiten wie auch die Software zu konzipieren.

Ein zentraler Erfolgsfaktor ist die User-Zentrierung: Die konsequente Ausrichtung der Applikation an den Bedürfnissen der Anwender steigert die Akzeptanz und fördert die aktive Nutzung. Eine benutzerfreundliche, moderne Lösung erleichtert nicht nur den Arbeitsalltag, sondern trägt auch massgeblich zur Effizienzsteigerung bei. Gleichzeitig müssen diese Anforderungen mit den organisatorischen und rechtlichen Vorgaben der Behörden in Einklang gebracht werden.

### 4.1.1 Anwender erwarten stabile, benutzerfreundliche Systeme, die ihre täglichen Arbeitsprozesse effektiv unterstützen

Um diese User-Zentrierung zu erreichen, ist es entscheidend, die tatsächlichen Anforderungen und Erwartungen der Nutzer zu verstehen und in den Entwicklungsprozess einfließen zu lassen. Nur so kann eine Fachapplikation entstehen, die den Arbeitsalltag der Beschäftigten optimal unterstützt und gleichzeitig die behördlichen Vorgaben erfüllt.

Beispiele für Erwartungen von Anwendern für eine nutzerorientierte Fachapplikation:

- Prozessnähe: Die Anwendung muss die realen Arbeitsabläufe der Justizbehörden abbilden und vereinfachen.  
*Beispiel:* Automatische Fristenerkennung in eingehenden Dokumenten oder intelligente Zuordnung von Akten zu Verfahren.
- Verfügbarkeit: Die Anwendung soll hochverfügbar erreichbar sein
- Performance: Kurze Ladezeiten und stabile Systemreaktionen, auch bei hoher Auslastung
- Benutzerfreundlichkeit: Eine klare, intuitive Oberfläche, die ohne lange Einarbeitung nutzbar ist
- Effizienzsteigerung: Unterstützung durch Automatisierung und intelligente Funktionen, um Routineaufgaben zu reduzieren
- Flexibilität: Anpassbarkeit an unterschiedliche Rollen, Aufgabenbereiche und zukünftige Anforderungen
- Barrierefreiheit: Unterstützung für unterschiedliche Nutzergruppen, zum Beispiel durch barrierefreie Oberflächen

### 4.1.2 Organisationen bringen Anforderungen mit, die jene der User ergänzen und Aspekte wie Governance, Sicherheit und Nachhaltigkeit sicherstellen

Neben den Erwartungen der Anwender spielen auch die strategischen und betrieblichen Anforderungen der Organisation eine zentrale Rolle. Diese ergänzen die Erwartungen der Nutzer und stellen sicher, dass die Fachapplikation nicht nur benutzerfreundlich, sondern auch wirtschaftlich, sicher und zukunftsfähig ist. Als mögliche Bedürfnisse der Organisationen wurden folgende identifiziert:

- Rechtssicherheit und Compliance: Einhaltung aller gesetzlichen Vorgaben, Datenschutz- und Sicherheitsstandards
- Standardisierung: Einheitliche Prozesse und Datenstrukturen zur Sicherstellung von Konsistenz und Effizienz, Integration justitia.swiss
- Nachhaltigkeit: Langfristige Nutzbarkeit und Weiterentwicklung ohne Stabilitätsverlust
- Kosten- und Ressourceneffizienz: Optimale Nutzung von Budget und IT-Ressourcen, inklusive Wartungsfreundlichkeit
- Interoperabilität: Nahtlose Integration in bestehende Systemlandschaften und Schnittstellen zu anderen Fachanwendungen

- Modularität: Anpassungsfähigkeit durch den gezielten modularen Aufbau der Software an unterschiedliche Fachverfahren und Organisationen
- Skalierbarkeit: Fähigkeit, zukünftige Anforderungen und steigende Nutzerzahlen ohne Leistungseinbußen zu bewältigen
- Transparenz und Nachvollziehbarkeit: Lückenlose Dokumentation und Auditierbarkeit aller Vorgänge
- Sicherheit: Schutz vor Cyberangriffen und Sicherstellung der Datenintegrität

#### 4.1.3 Durch den Einbezug der Justizbehörden werden mögliche Bedürfnisse zu konkreten Anforderungen

Um die richtigen architekturbezogenen und technischen Entscheidungen zu treffen, wurden strukturierte Workshops und Interviews mit verschiedenen Justizbehörden durchgeführt. Ziel war es, die Bedürfnisse zu konkretisieren und ein umfassendes Bild der aktuellen Herausforderungen, Erwartungen und Prioritäten zu gewinnen.

Im Rahmen der Workshops und Interviews wurde deutlich, dass sich die im Vorfeld identifizierten Bedürfnisse in konkrete, bewertbare Anforderungen überführen lassen. Um eine klare Entscheidungsgrundlage für Architektur und Betrieb zu schaffen, wurden diese Anforderungen in zwei Kategorien unterteilt:

- Muss-Anforderungen: Diese sind unverzichtbar, um die rechtlichen, sicherheitsrelevanten und funktionalen Mindeststandards zu erfüllen. Sie bilden die Grundlage für den Betrieb einer modernen Fachapplikation im Justizumfeld.
- Kann-Anforderungen: Diese sind wünschenswert und erhöhen den Nutzen, die Flexibilität oder die Zukunftsfähigkeit der Lösung, sind jedoch nicht zwingend für die Grundfunktionalität erforderlich.

Diese Differenzierung ermöglicht es, kritische Anforderungen verbindlich umzusetzen, während gleichzeitig Gestaltungsspielräume für Innovation und Optimierung erhalten bleiben. Sie dient zudem als zukünftige Basis für die Priorisierung in Architekturentscheidungen.

Im Folgenden werden zunächst die in den Workshops identifizierten Muss-Anforderungen dargestellt.

#### Technische und funktionale Anforderungen

- Single-Source-of-Truth: Keine redundanten Daten im System
- Mandantenfähigkeit: Strikte Trennung der Daten und Zugriffe (zum Beispiel über separate Datenbankschemata)
- Hohe Performance: Stabile Reaktionszeiten auch bei hoher Last
- Hohe Verfügbarkeit: Minimierung von Ausfallzeiten, definierte Wartungsfenster
- Kompatibilität: Browserbasierte Nutzung auf verschiedenen Endgeräten
- Backup & Recovery: Regelmässige Sicherungen und getestete Wiederherstellung
- Wartbarkeit: Klare Update-Strategien, Monitoring und Diagnosemöglichkeiten
- Intelligente Suche: Effiziente, fehlertolerante Suche in grossen Datenbeständen

#### Sicherheits- und Compliance-Anforderungen

Eine zentrale Anforderung ist die verschlüsselte Kommunikation, die den Schutz sensibler Daten sicherstellt. Hier kommt das Protokoll TLS (Transport Layer Security) zum Einsatz. TLS ist ein etablierter Sicherheitsstandard, der die Datenübertragung zwischen Systemen durch starke Verschlüsselung absichert. Man kann sich TLS wie einen digitalen Umschlag vorstellen: Der Inhalt einer Nachricht wird so verpackt, dass er unterwegs weder geöffnet noch verändert werden kann. Nur der vorgesehene Empfänger ist in der Lage, die Nachricht zu lesen und ihre Echtheit zu prüfen.

TLS schützt dabei vor:

- Man-in-the-Middle-Angriffen: Abfangen und Manipulieren von Daten wird verhindert
- Datendiebstahl: Verschlüsselte Informationen sind für Dritte nicht lesbar
- Datenmanipulation: Integritätsprüfungen verhindern unbemerkte Änderungen
- Identitätsmissbrauch: Authentifizierung stellt sicher, dass Kommunikationspartner echt sind

TLS bildet dabei die Grundlage für sichere Webkommunikation (z. B. HTTPS) und ist unverzichtbar für die Einhaltung von Datenschutzvorgaben sowie für den Schutz vor Cyberangriffen.

#### Weitere wesentliche Anforderungen sind

- Revisionssicherheit: Vollständige Nachvollziehbarkeit aller Zustände
- Single-Sign-On (SSO): Einheitliche Authentifizierung über bestehende Identitätslösungen
- Datenschutz & Zugriffstrennung: Auditierbare Trennung von Daten und Rechten auf allen Ebenen

#### Betriebs- und Kostenanforderungen

- Total Cost of Ownership: Ganzheitliche Kostenbetrachtung, im Zweifel Fokus auf Betriebskosten
- Betriebsmodell: Betrieb vor Ort wird grundsätzlich bevorzugt; Private Cloud für manche Behörden möglich; Public Cloud wird abgelehnt
- Technologie-Kenntnisse: Kenntnisse in den Betriebsorganisationen sollen beachtet werden um diese nicht zu überfordern.

Neben den unverzichtbaren Muss-Anforderungen gibt es eine Reihe von wünschenswerten Funktionen, die den Nutzen und die Zukunftsfähigkeit der Fachapplikation erhöhen können:

- Reduktion von Abhängigkeiten zwischen Software-Modulen
- Open-Source-Komponenten
- Unabhängigkeit von einem Dienstleister
- Möglichkeit der Integration von KI: Offenheit für KI-basierte Funktionen

#### 4.1.4 Funktionsgruppen Landkarte

Neben den technischen und nicht-funktionalen Anforderungen muss eine Fachapplikation auch funktional alle Bedürfnisse abdecken. Dafür wurden mit verschiedenen Behörden Interviews geführt. Die folgende Funktionsgruppen Landkarte zeigt das Ergebnis dieser Interviews übersichtlich, welche zentralen Funktionen und Querschnittsfähigkeiten eine moderne Fachanwendung im Justizbereich abdecken muss.



Abbildung 4 Funktionsgruppen Landkarte Fachapplikation

Die Funktionen lassen sich grundsätzlich in die folgenden Bereiche einteilen:

- Fallbearbeitung
- Digitale Akte
- Administration
- Kommunikation
- Dokumente
- Fallbuchhaltung
- Analyse / Reporting

Die Grafik beschreibt dabei die darunter liegenden Funktionalitäten und zeigt, wie diese die fachliche Breite einer modernen Fachapplikation abdecken<sup>1</sup>. Somit ist mit der Implementierung dieser Funktionalitäten sichergestellt, dass alle relevanten Funktionsbereiche abgedeckt werden und die Applikation die Prozesse der gesamten Strafjustizkette unterstützen kann.

#### 4.2 Hohe Anforderungen an eine Fachapplikation der Zukunft

Die Analyse der Bedürfnisse und Anforderungen zeigt deutlich, dass eine moderne Fachapplikation für die Justizbehörden sowohl den hohen Erwartungen der Anwender als auch den strategischen und betrieblichen Anforderungen der Organisationen gerecht werden muss. Die konsequente Ausrichtung auf die Nutzerperspektive ist dabei ein zentraler Erfolgsfaktor, da sie die Akzeptanz fördert, die Nutzung erleichtert und die Effizienz im Arbeitsalltag steigert.

Gleichzeitig wurde in den Workshops und Interviews deutlich, dass es nicht die eine ideale Lösung für alle gibt. Vielmehr müssen technologische Möglichkeiten und innovative Ansätze in einem ausgewogenen Verhältnis zur Umsetzbarkeit in den jeweiligen Betriebsorganisationen stehen. Es ist wichtig, differenzierte Lösungen zu entwickeln, die den unterschiedlichen Reifegraden, Ressourcen und Kenntnisständen der Organisationen gerecht werden.

Wesentliche Erkenntnisse im Überblick:

- Die konsequente Ausrichtung an den Bedürfnissen der Anwender ist entscheidend, muss jedoch mit der betrieblichen Umsetzbarkeit in Einklang stehen.
- Technologische Innovationen wie Automatisierung oder künstliche Intelligenz bieten grosses Potenzial, sollten jedoch schrittweise und bedarfsgerecht eingeführt werden.
- Standardisierte und modulare Architekturen ermöglichen eine flexible Anpassung an unterschiedliche Organisationen und Fachverfahren.
- Technologische Entscheidungen und Betriebsmodelle müssen die vorhandenen Kompetenzen und Ressourcen der Betriebsorganisationen berücksichtigen.
- Ein gestuftes Vorgehen schafft die Möglichkeit, ambitionierte Ziele zu verfolgen, ohne bestehende Strukturen zu überfordern.
- Für die Organisationen sind die Einhaltung und Umsetzung gesetzlicher Anforderungen ein zentrales Kriterium.

Diese Erkenntnisse bilden die Grundlage für eine nachhaltige, tragfähige und zukunftsorientierte Fachapplikation, die sowohl den Anforderungen der Justiz als auch den Rahmenbedingungen der Organisationen gerecht wird. Dabei ist zu berücksichtigen, dass zwar Themen wie Public Cloud und Private Cloud an Relevanz gewinnen, in der Praxis jedoch der Betrieb im eigenen Rechenzentrum weiterhin bevorzugt wird. Aus diesem Grund muss die Architektur so gestaltet sein, dass unterschiedliche Betriebsformen möglich sind und diese Offenheit auch in den Konzepten vorgesehen ist. Damit bleibt die Wahl flexibel, ob die Lösung im eigenen Rechenzentrum, in einer Private Cloud oder in einer Public Cloud betrieben wird.

## 5 Zielarchitektur

Dieses Kapitel beschreibt die Zielarchitektur einer modernen Fachapplikation für die Justiz. Es führt von den grundlegenden Architekturprinzipien über die technischen und organisatorischen Anforderungen bis hin zur praktischen Umsetzung. Dabei werden zunächst die Grenzen heutiger monolithischer Systeme aufgezeigt und daraus die zentralen Grundsätze der künftigen Architektur abgeleitet. Anschliessend werden

---

<sup>1</sup> Eine detaillierte Beschreibung der einzelnen Funktionen befindet sich im externen Dokument «2025-10-22 - Funktionalen Anforderungen HIS.xlsx».

die modularen, ereignisorientierten und revisionssicheren Prinzipien im Detail erläutert, bevor anhand eines exemplarischen Prozesses gezeigt wird, wie diese Architektur in der Praxis funktioniert. Das Kapitel schliesst mit der Beschreibung zentraler Komponenten wie Eventhub, Eventstore und Adapter sowie einer Einordnung ihrer fachlichen und technischen Rollen

## 5.1 Grundsätze der künftigen Architektur

Die Analyse der Bedürfnisse und Anforderungen hat gezeigt, dass die Erwartungen an eine moderne Fachapplikation hoch und vielfältig sind. Anwender wünschen sich intuitive und verlässliche Werkzeuge. Organisationen verlangen Rechtssicherheit, Effizienz und Zukunftsfähigkeit.

Es entsteht ein Spannungsfeld (siehe Abbildung 5 Spannungsfeld Anforderungen Architektur). Einerseits müssen bewährte Arbeitsabläufe unterstützt und rechtliche Vorgaben lückenlos erfüllt werden. Andererseits braucht es Freiräume für Innovation, Automatisierung und den Einsatz neuer Technologien. Eine Architektur, die diesen Spagat nicht leisten kann, wird unweigerlich zu einem Hemmschuh für die Weiterentwicklung der Justiz.

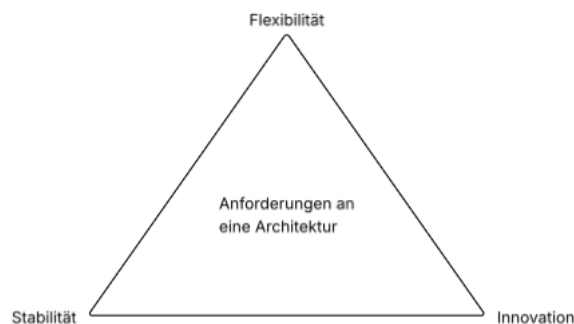


Abbildung 5 Spannungsfeld Anforderungen Architektur

Genau hier setzt die Zielarchitektur an. Sie bildet die Brücke zwischen den heutigen Herausforderungen und den zukünftigen Möglichkeiten. Sie zeigt einen Weg auf, wie sich Benutzerfreundlichkeit und Effizienz mit Stabilität und Sicherheit vereinen lassen. Vor allem aber eröffnet sie den Justizbehörden die Chance, ihre Verfahren modular, flexibel und nachhaltig weiterzuentwickeln. Dies gelingt nur ohne die Last monolithischer Strukturen. In diesem Kapitel werden zunächst die Herausforderungen der aktuellen monolithischen Architektur dargestellt. Anschliessend werden neue Architekturprinzipien und Methodiken erläutert, die notwendig sind, um im weiteren Verlauf des Kapitels die Ausgestaltung der Software zu präzisieren.

### 5.1.1 Anforderungen zeigen Grenzen monolithischer Architekturen in aktuellen Applikationen auf

Die gegenwärtig eingesetzten Fachapplikationen der Justiz basieren überwiegend auf monolithischen Softwarearchitekturen. In diesen Systemen sind sämtliche Funktionalitäten eng miteinander verknüpft, angefangen bei der Benutzeroberfläche über die Geschäftslogik bis hin zur Datenhaltung (Abbildung 6 Monolithische Architektur)

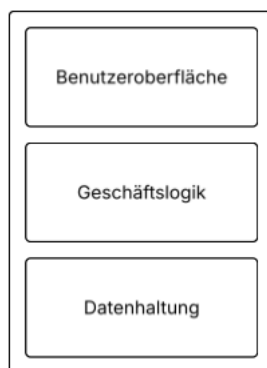


Abbildung 6 Monolithische Architektur

Dies bringt einige Vorteile mit sich:

- Schneller Start: Für kleine Teams oder Prototypen ist ein Monolith am Anfang oft einfacher und schneller umzusetzen
- Geringere Anfangskosten: Weniger benötigte Infrastruktur sowie einfachere Installation und Konfiguration reduzieren den initialen Aufwand

Die Vorteile drehen sich aber über längere Entwicklungs- und Betriebszeiten ins Negative. Es herrscht eine enge Verzahnung der einzelnen Softwareteile, sowohl zwischen der Benutzeroberfläche, der Geschäftslogik oder Datenhaltung als auch innerhalb der einzelnen Teile.

Als Folge davon ist jede Weiterentwicklung komplex und aufwändig. Zudem wird die Abhängigkeiten zwischen den Komponenten erhöht.

Dies führt zu erheblichen technischen und organisatorischen Nachteilen:

- Hohe Komplexität: Änderungen an einer einzelnen Komponente erfordern ein tiefes Verständnis des gesamten Systems.
- Risiko von Seiteneffekten: Anpassungen können mitunter unvorhersehbare und unbeabsichtigte Auswirkungen haben.
- Lange Entwicklungszyklen: Umfangreiche Tests, die über das gesamte System notwendig werden, verlängern die Bereitstellungszeiten für einen neuen Release.

Aus diesen technischen und organisatorischen Nachteilen ergeben sich u.a. folgende Konsequenzen:

- Hohe Wartungs- und Weiterentwicklungskosten
- Hohe Eintrittshürden für neue Entwicklungsdienstleister
- Abhängigkeit von wenigen Know-how-Trägern

In einem Umfeld, in dem rechtliche, organisatorische und technologische Anforderungen laufend angepasst werden müssen, erweist sich diese Architekturform als nicht mehr tragfähig. Die heutigen monolithischen Systeme bremsen die digitale Transformation der Justiz und verhindern die notwendige Agilität.

### 5.1.2 Modularisierung als strategische Antwort

Die Probleme monolithischer Architekturen sind in der Informatik seit langem bekannt. Als Antwort bietet sich die konsequente Trennung von Funktionseinheiten in eigenständige, lose gekoppelte Module an. Dies ist ein grundlegender Wandel gegenüber früheren, eng verzahnten Softwareteilen.

Moderne Architekturen, insbesondere im Kontext von Microservices, segmentieren Anwendungen entlang fachlicher Domänen. Die Kommunikation erfolgt über klar definierte Schnittstellen, um die funktionale Kopplung sicherzustellen. So ist gewährleistet, dass die einzelnen Module miteinander kommunizieren und bei Bedarf ausgetauscht werden können. Abbildung 7 zeigt das Prinzip der Modularisierung auf einen Blick.

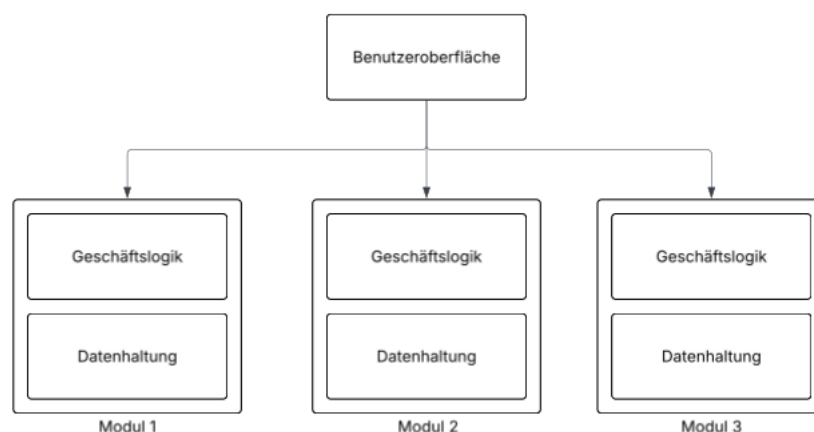


Abbildung 7 Modularisierte Anwendung mit getrennten Modulen für Geschäftslogik und Datenhaltung sowie separater Benutzeroberfläche



Wie in Abbildung 7 erkennbar, verfügt jedes Modul über seine eigene Geschäftslogik und die dazugehörige Datenhaltung. Die Benutzeroberfläche greift ausschliesslich über definierte Schnittstellen zu. Für die Praxis bedeutet das, dass Weiterentwicklungen und Korrekturen gezielt in einzelnen Modulen erfolgen können, ohne das Gesamtsystem zu beeinträchtigen. Störungen bleiben lokal und lassen sich beheben, während andere Funktionen weiterlaufen können. Es entstehen klare Zuständigkeiten und definierte Datenhoheit je Modul. Dienstleister können einzelne Komponenten unabhängig entwickeln und integrieren, da nur die Schnittstellen bekannt sein müssen und nicht die gesamte Systemlogik. Zudem wird es damit möglich, Dienstleister ohne tiefes Domänenwissen mit der Entwicklung spezifischer Komponenten zu beauftragen. Damit kann die modulare Architektur die Eintrittsbarrieren für neue Marktteilnehmer senken.

### 5.1.3 Die richtige Segmentierung als Schlüssel

Neben den klaren Vorteilen der Modularisierung gibt es auch Herausforderungen, die in dieser Architektur bewältigt werden müssen. Die Aufteilung einer Applikation in eigenständige Module ist dabei eine anspruchsvolle Herausforderung. Dies ist nicht nur eine technische Entscheidung, sondern ein zentrales Qualitätskriterium für eine gelungene Modularisierung. Eine falsche Segmentierung führt schnell zu neuen Abhängigkeiten, redundanten Datenmodellen und komplexen Integrationsszenarien und damit zu denselben Problemen, die man eigentlich vermeiden wollte.

Hier setzt das Konzept des Domain-driven Design (DDD) an. Das Konzept des DDD hilft dabei, komplexe Softwaresysteme besser zu strukturieren, indem es sich an der realen Arbeitswelt orientiert. Es verfolgt das Ziel, Fachlichkeit und Technik enger miteinander zu verbinden, sodass Entwickler und Fachleute eine gemeinsame Sprache sprechen. Die sogenannte Domäne beschreibt dabei den Themen oder Aufgabenbereich, um den es in der Software geht, zum Beispiel Justiz, Gesundheitswesen oder Finanzen. In dieser Studie bezieht sich die Domäne auf den Bereich der Justiz. DDD teilt grosse Systeme in klar abgegrenzte Teile, sogenannte Bounded Contexts. Jeder dieser Bereiche funktioniert wie ein eigenes kleines System. Er hat seine eigenen Regeln, seine eigenen Daten und klar definierte Verbindungen zu anderen Bereichen. Auf diese Weise bleibt die Software übersichtlich, verständlich und kann gezielt erweitert oder angepasst werden, ohne dass andere Teile des Systems unnötig beeinflusst werden.

Die Grundprinzipien von DDD:

- **Fokus auf die Fachlogik**  
Die Software soll die fachlichen Regeln und Abläufe eines Arbeitsbereichs korrekt abbilden. Entscheidend ist, dass nicht allein technische Funktionen umgesetzt werden, sondern dass das zugrunde liegende Fachwissen präzise in die Anwendung übertragen wird. Dabei muss sich die fachliche Logik auch im Quellcode widerspiegeln, es reicht also nicht, dass nur das Ergebnis stimmt.  
Im Justizbereich bedeutet das, dass Fristen exakt nach den gesetzlichen Vorgaben berechnet werden.
- **Bounded Context**  
Ein Bounded Context ist ein klar abgegrenzter Bereich innerhalb einer Domäne. Er umfasst einen Teil der fachlichen Aufgaben und bildet diesen mit eigenen Begriffen, Regeln und Datenstrukturen ab. Innerhalb eines solchen Kontexts haben Begriffe eine eindeutige Bedeutung, die nur dort gilt. Jeder Kontext verfügt über ein eigenes Datenmodell und eine eigene Logik, um seinen Ausschnitt der Domäne präzise zu beschreiben.  
Im Justizbereich gibt es zum Beispiel einen Kontext für die Dossierverwaltung und einen weiteren für das Dokumentenmanagement. Beide gehören zur Domäne der Justiz, arbeiten jedoch unabhängig voneinander und kommunizieren über definierte Schnittstellen.
- **Strategisches Design**  
Das strategische Design beschreibt, wie die verschiedenen Bounded Contexts einer Domäne miteinander in Beziehung stehen. Es legt fest, wie die Grenzen zwischen den einzelnen Bereichen verlaufen und wie sie zusammenwirken, damit die gesamte Domäne als funktionierendes Ganzes abgebildet wird. Ziel ist es, Abhängigkeiten zu vermeiden und eine lose Kopplung zu erreichen. Die Kontexte sollen miteinander kommunizieren können, ohne sich gegenseitig zu beeinflussen.  
Um dies zu gewährleisten, interagieren die verschiedenen Bounded Contexts über klar definierte Schnittstellen, anstatt direkt auf die Daten anderer Bereiche zuzugreifen oder diese zu verändern.
- **Taktisches Design**  
Das taktische Design beschreibt den inneren Aufbau eines Bounded Contexts und legt fest, wie die

fachliche Logik im Programmcode umgesetzt wird. Dafür werden Bausteine verwendet, mit denen sich die fachlichen Konzepte im Code abbilden lassen.

- Entity: Dies beschreibt ein fachliches Objekt, das über längere Zeit bestehen bleibt, mit einer eindeutigen Identität.
- Value Object: Ein Wert ohne eigene Identität. Es wird allein durch seine Eigenschaften bestimmt. Zwei Value Objects gelten als gleich, wenn sie denselben Wert haben.
- Aggregate: Ein Aggregat fasst mehrere zusammengehörende Entities und Value Objects zu einer Einheit zusammen, die als Ganzes verwaltet werden.
- Repository: Es organisiert den Zugriff auf Entities oder Aggregate. Es trennt die Fachlogik von technischen Details wie der Datenspeicherung und sorgt so für eine saubere Strukturierung des Codes.

Wie diese Bausteine zusammenspielen, zeigt das folgende Beispiel: Im Bounded Context „Dossierverwaltung“ gibt es eine Entity „Fall“ mit einer eindeutigen Fallnummer. Innerhalb dieser Entity werden Value Objects wie „Adresse“ und „Datum“ verwendet. Der Fall mit allen zugehörigen Dokumenten bildet ein Aggregat. Das Repository verwaltet schliesslich den Zugriff auf die unterschiedlichen Fälle und ihre Dokumente.

Durch den Einsatz von DDD kann eine Software in einzelne, unabhängig voneinander arbeitende Module aufgeteilt werden. Die Methodik bildet damit das Rückgrat des zentralen Architekturprinzips der Modularisierung.

Darüber hinaus schafft DDD durch die Verwendung klar definierter Fachbegriffe eine gemeinsame Sprache zwischen Fachspezialisten und Entwicklern. Diese gemeinsame Terminologie fördert das gegenseitige Verständnis und verhindert Missverständnisse in der fachlichen und technischen Kommunikation. Auf dieser Grundlage kann ein Glossar entstehen, das die zentralen Begriffe der Domäne dokumentiert und als verbindliche Referenz dient. So wird gewährleistet, dass Fachlichkeit und Technik dauerhaft im Einklang bleiben und sich die Weiterentwicklung der Systeme an einem gemeinsamen Begriffsverständnis orientiert.

#### 5.1.4 Ereignisorientierung als architektonisches Ordnungsprinzip

Die zunehmende Modularisierung von Fachanwendungen führt zu einer höheren Komplexität in der Koordination zwischen den Modulen. Klassische Systeme lösen dieses Problem häufig durch Orchestrierung: Eine zentrale Workflow-Engine steuert die Abläufe Schritt für Schritt und legt fest, wie ein Vorgang verarbeitet wird. Dieses Vorgehen bietet zwar Transparenz und Kontrolle, es schafft jedoch auch enge Abhängigkeiten. Jede Anpassung erfordert Änderungen in der zentralen Steuerungslogik. Fehler in der Steuerung können zudem den gesamten Prozess blockieren.

Im Gegensatz dazu basiert die ereignisorientierte Architektur (EDA) auf einem dezentralen Modell. Sie geht davon aus, dass Geschäftsprozesse durch Ereignisse und Tätigkeiten beschrieben werden können. Ereignisse kennzeichnen dabei Zustände oder Auslöser, die einen Prozess anstossen oder beenden, zum Beispiel wenn ein Antrag eingeht oder ein Urteil zugestellt wird. Die beteiligten Akteure innerhalb eines Geschäftsprozesses reagieren selbstständig auf die für sie relevanten Ereignisse und führen Tätigkeiten aus. Diese Tätigkeiten erzeugen wiederum neue Ereignisse, sodass der Geschäftsprozess als eine Abfolge von Zustandsänderungen beschrieben werden kann.

Dieses Prinzip überträgt EDA in die Softwarearchitektur. Die unterschiedlichen Module eines Softwaresystems sollen eigenständig auf die für sie relevanten Ereignisse handeln. Dafür werden alle Ereignisse auf einem zentralen Dienst veröffentlicht. Dieser Dienst wird Eventhub genannt. Seine Rolle ist die des Vermittlers: Er empfängt Ereignisse und verteilt sie an alle betroffenen bzw. relevanten Module. Er bestimmt dabei nicht selbst, welches Modul ein Ereignis zugestellt bekommt. Ein Modul abonniert beim Start des Moduls die gewünschten Ereignisse beim Eventhub. Im Anschluss bekommt es die jeweiligen Ereignisse automatisch zugestellt. Eine zentrale Steuerung durch den Eventhub findet nicht statt. Dadurch erfordert die Anpassung eines Ereignisses keine Änderungen an einer zentralen Steuerung.

Obwohl der Eventhub keine Steuerungsfunktion übernimmt, ist seine Verfügbarkeit entscheidend für die Funktionsfähigkeit des Gesamtsystems. Als zentrales Kommunikationssystem muss er daher redundant ausgelegt sein, um bei einem Ausfall die kontinuierliche Verarbeitung von Ereignissen sicherzustellen und Unterbrechungen in den Geschäftsprozessen zu vermeiden.



Im direkten Vergleich der Architekturtypen ergeben sich deutliche Unterschiede:

| Merkmal               | Orchestrierung (zentral gesteuert)                | Ereignisorientierung (EDA)   |
|-----------------------|---|--|
| Steuerung             | Zentrale Engine legt Abläufe fest                 | Jedes Modul reagiert selbstständig   |
| Kopplung              | Enge Abhängigkeiten von der zentralen Logik       | Lose Kopplung über Ereignisse  |
| Erweiterbarkeit       | Neue Funktionen nur durch Anpassung der Steuerung | Neue Module können Ereignisse abonnieren, ohne bestehende Module zu ändern |
| Fehlerrobustheit      | Fehler wirken oft auf den gesamten Prozess        | Fehler bleiben lokal begrenzt  |
| Skalierbarkeit        | Schwer unabhängig skalierbar                      | Module können gezielt skaliert werden                                      |
| Exemplarischer Aufbau |   |  |

Pfeile zeigen die Kommunikations- bzw. Steuerungsrichtung

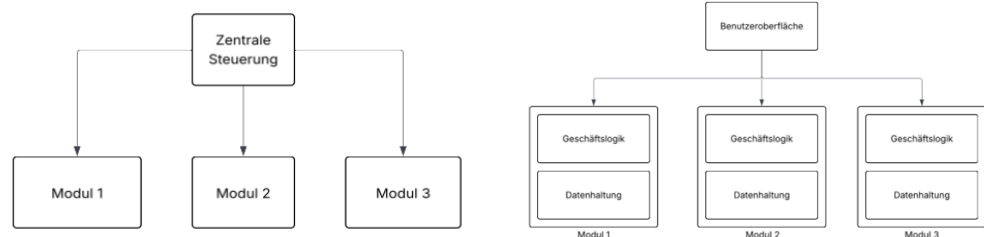


Tabelle 1 Vergleich Architektur zentrale Orchestrierung und Ereignisorientierung

Für die Justiz bietet eine ereignisgesteuerte Architektur entscheidende Vorteile. Sie ermöglicht eine hohe Modularität, da Module unabhängig voneinander entwickelt, erweitert oder ersetzt werden können. Wird beispielsweise das Modul für den elektronischen Versand von Urteilen modernisiert, bleiben andere Komponenten wie Register, Fristenkontrolle oder Aktenführung unverändert. Die lose Kopplung reduziert Abhängigkeiten und erleichtert Anpassungen. Neue Vorgaben wie etwa zur Akteneinsicht lassen sich in einem eigenen Modul umsetzen, ohne dass bestehende Funktionen verändert werden müssen.

Auch die Auswirkungen eines Ausfalls werden verringert: Wenn die Dokumentensuche nicht funktioniert, laufen Fristenüberwachung und Urteilsversand dennoch weiter. Das Gesamtsystem bleibt stabil, selbst wenn einzelne Teile vorübergehend nicht verfügbar sind. Technisch setzt dies eine zuverlässige Ereignisverarbeitung voraus. Der zentrale Eventhub muss redundant ausgelegt sein, da er als Kommunikationsdrehscheibe fungiert. So wird sichergestellt, dass die Ereignisverarbeitung auch bei Ausfällen fortgeführt werden kann.

Darüber hinaus entsteht ein hoher Investitionsschutz, da bestehende Systeme nicht vollständig ersetzt, sondern schrittweise modernisiert werden. Bei Bedarf lassen sich einzelne Module gezielt austauschen: Fällt etwa das Zustellungsmodul aus, kann es ersetzt werden, während die Aktenverwaltung weiter genutzt wird. Für Mitarbeitende ist das ein grosser Vorteil, da sie mit vertrauten Werkzeugen weiterarbeiten können.

Innovationen wie KI-gestützte Dokumentenanalyse lassen sich einfach integrieren. Ein Modul kann beispielsweise automatisch Urteile vorschlagworten, sobald das Ereignis „Urteil erstellt“ eintritt, ohne dass das bestehende Modul angepasst werden muss. Besonders rechenintensive Funktionen wie die Suche können separat skaliert werden. So belastet eine erhöhte Nachfrage nur den betroffenen Teil, nicht das gesamte System.

EDA bietet damit die notwendige Flexibilität und Zukunftsfähigkeit, um eine moderne Fachapplikation modular und langfristig weiterzuentwickeln.

### 5.1.5 Event Sourcing: Revisionssicherheit durch vollständige Ereignisprotokollierung

Die ereignisgesteuerte Architektur ermöglicht Flexibilität, indem Module eigenständig auf Ereignisse reagieren. Für die Justiz reicht es jedoch nicht aus, Ereignisse nur im Moment zu verarbeiten. Sie müssen auch dauerhaft nachvollziehbar und rechtssicher dokumentiert werden. Genau hier setzt Event Sourcing an. Dabei wird nicht der aktuelle Zustand eines Objekts gespeichert, sondern die gesamte Historie aller zustandsverändernden Ereignisse. Der aktuelle Zustand ergibt sich durch die chronologische Verarbeitung dieser Ereignisse.

Für die Justiz bedeutet dies, dass jeder Schritt in einem Verfahren, etwa die Einreichung eines Antrags, die Erstellung eines Urteils oder die Zustellung eines Bescheids, revisionssicher dokumentiert wird. So entsteht ein vollständiges digitales Protokoll, das Transparenz und Rechtssicherheit gewährleistet und auch Jahre später noch eine verlässliche Rekonstruktion ermöglicht.

Technisch wird dies dadurch erreicht, dass alle Ereignisse unveränderlich in einem Eventstore gespeichert werden. Ein Eventstore ist eine Datenbank, die nicht nur Ergebnisse speichert, sondern die gesamte Abfolge von Ereignissen. Neue Module können diese Ereignisse nutzen, ohne bestehende Systeme verändern zu müssen, und bestehende Abläufe lassen sich durch das erneute Abspielen der gespeicherten Ereignisse zuverlässig rekonstruieren. Dieses erneute Abspielen bedeutet, dass die gesamte Ereignisfolge noch einmal durchlaufen wird, um den Zustand wiederherzustellen (siehe Abbildung 8 Exemplarischer Aufbau einer EDA-Architektur mit Eventstore).

Da der Eventstore die Grundlage für die Rekonstruktion und Nachvollziehbarkeit bildet, ist er eine kritische Komponente. Ein Ausfall würde die Rechtssicherheit und die Wiederherstellbarkeit des Systems gefährden. Deshalb muss der Eventstore redundant ausgelegt werden um eine hohe Verfügbarkeit und Ausfallsicherheit zu gewährleisten. Zusätzlich sollten regelmässige Backups erstellt und an getrennten Speicherorten aufbewahrt werden. Diese Backups müssen auf Integrität geprüft und im Ernstfall schnell wiederherstellbar sein, um auch bei schwerwiegenden Fehlern oder Datenkorruption die Ereignishistorie zu sichern.

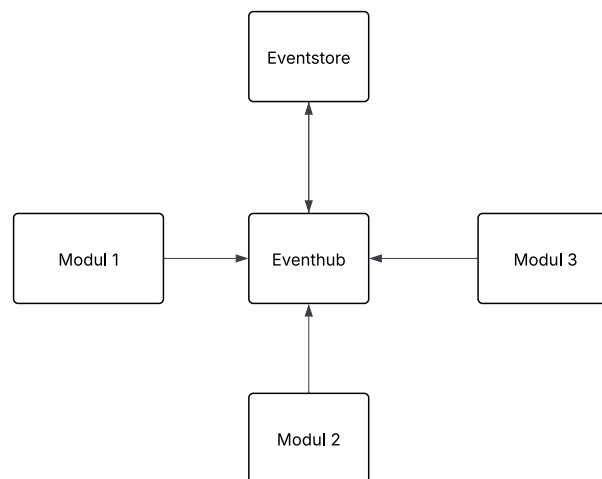


Abbildung 8 Exemplarischer Aufbau einer EDA-Architektur mit Eventstore

Event Sourcing ergänzt damit die Architekturprinzipien Modularisierung und Ereignisorientierung durch ein starkes Muster, das Nachvollziehbarkeit und Revisionssicherheit sicherstellt. Gerade im justiziellen Umfeld ist dies unverzichtbar, da rechtliche Vorgänge jederzeit lückenlos dokumentiert und überprüft werden müssen.

### 5.1.6 Dokumentation und Standardisierung als Voraussetzung für Interoperabilität

Damit die Vorteile der Modularisierung jedoch in der Praxis wirksam werden, genügt es nicht, Ereignisse nur zu speichern oder zu verteilen. Ebenso wichtig ist, dass die Art und Weise, wie Ereignisse beschrieben und Schnittstellen gestaltet werden, einheitlich geregelt ist. Erst durch konsequente Dokumentation und Standardisierung entsteht Interoperabilität, also die Fähigkeit unterschiedlicher Systeme und Module, zuverlässig miteinander zu arbeiten. Nur wenn klar definiert ist, wie Daten strukturiert sind und welche Begriffe in welchem Zusammenhang verwendet werden, können Module unterschiedlicher Hersteller und Behörden nahtlos zusammenarbeiten. Diese Fähigkeit nennt man Interoperabilität. Sie beschreibt, dass Systeme trotz unterschiedlicher Herkunft zuverlässig miteinander Informationen austauschen können. Für synchrone Schnittstellen, also den direkten Austausch von Informationen zwischen zwei Modulen in Echtzeit, bietet sich der Einsatz von OpenAPI an. Eine Schnittstelle ist dabei eine Art Vertrag, der festlegt, wie zwei Systeme miteinander sprechen dürfen. OpenAPI ist ein international verbreiteter Standard, mit dem solche Schnittstellen genau beschrieben werden. Man kann es sich wie einen sehr präzisen Bauplan vorstellen, der sowohl für Entwickler als auch für Prüfer verständlich ist.

Für asynchrone Schnittstellen, also wenn Ereignisse zeitlich unabhängig verschickt und verarbeitet werden, eignet sich AsyncAPI. Während bei synchronen Schnittstellen eine sofortige Antwort erwartet wird, ist es bei asynchronen Schnittstellen so, als würde man eine Nachricht in ein Postfach legen. Das

empfangende Modul holt die Nachricht ab, wenn es dazu bereit ist. AsyncAPI beschreibt genau, wie diese Nachrichten aussehen müssen, damit sie von allen Modulen verstanden werden können.

OpenAPI für synchrone und AsyncAPI für asynchrone Kommunikation haben einen grossen praktischen Vorteil. Sie ermöglichen nicht nur eine klare Dokumentation, sondern können auch automatisch technische Hilfsmittel erzeugen, etwa Programmcodes für Entwickler, Tests zur Qualitätssicherung oder Bausteine für die Integration zwischen Modulen. Dadurch werden Entwicklungszyklen verkürzt, Fehlerquellen reduziert und die Wirtschaftlichkeit der Lösung gesteigert.

Darüber hinaus ist es für die Justiz entscheidend, sich an bestehenden Normen zu orientieren. Ein Beispiel sind die Schweizer Standards eCH-0051 und eCH-0280. Diese Normen beschreiben verbindlich, wie bestimmte Informationen in der öffentlichen Verwaltung ausgetauscht werden sollen, zum Beispiel welche Felder in einem elektronischen Dokument enthalten sein müssen. Werden solche Standards konsequent genutzt, können Investitionen geschützt, Abhängigkeiten von einzelnen Herstellern oder Personen verringert und die langfristige Interoperabilität gewährleistet werden.

### 5.1.7 Zusammenfassung

Die Einführung von Standards wie OpenAPI, AsyncAPI sowie bewährten Normen wie eCH-0051 und künftigen Normen der eCH-Fachgruppe Polizei/Justiz macht deutlich, dass technische Modularisierung nur dann ihr volles Potenzial entfalten kann, wenn Schnittstellen und Ereignisse konsequent dokumentiert und vereinheitlicht werden. Erst die Kombination aus modularen Bausteinen, einer flexiblen ereignisorientierten Steuerung, einer revisionssicheren Speicherung von Ereignissen und verbindlichen Standards schafft die Grundlage für eine belastbare und zukunftsfähige Architektur.

Dies bedeutet, dass eine moderne Fachapplikation auf zwei tragende Architekturprinzipien aufbauen sollte: Modularisierung sorgt dafür, dass Verfahren in klar abgegrenzte, unabhängige Module zerlegt werden können. Ereignisorientierung stellt sicher, dass diese Module flexibel und robust miteinander interagieren. Ergänzend dazu bietet Event Sourcing die notwendige Nachvollziehbarkeit und Rechtssicherheit, während Standardisierung die Interoperabilität zwischen unterschiedlichen Modulen, Behörden und Partnern garantiert.

Durch diese Kombination kann eine Fachapplikation entwickelt werden, die nicht nur den heutigen Anforderungen genügt, sondern auch auf künftige Entwicklungen vorbereitet ist. Sie vereint Benutzerfreundlichkeit mit Stabilität, schafft Spielräume für Innovation und erfüllt zugleich die hohen Ansprüche an Rechtssicherheit und Transparenz. Im nächsten Schritt wird daher der schematische Aufbau der Zielarchitektur beschrieben. Dieser Aufbau zeigt, wie diese Prinzipien in eine konkrete Struktur überführt werden können.

## 5.2 Schematischer Aufbau der Zielarchitektur einer modernen Fachapplikation

Die zuvor dargestellten Prinzipien Modularisierung, Ereignisorientierung, Event Sourcing und Standardisierung bilden das Fundament für eine zukunftsgerichtete Fachapplikation. Damit diese Prinzipien nicht abstrakt bleiben, sondern im Alltag wirken, müssen sie in eine klare technische Struktur überführt werden.

Diese Struktur beschreibt, welche Bausteine für den Betrieb einer modernen Fachapplikation notwendig sind, wie diese miteinander verbunden sind und welche Aufgaben sie jeweils übernehmen.

### 5.2.1 High-Level-Architektur und Komponentenüberblick

Die Zielarchitektur gliedert sich in drei zentrale Komponenten: Kernapplikation, Eventhub sowie Breeds und Adapter. Gemeinsam bilden sie ein System, das Stabilität, Flexibilität, Nachvollziehbarkeit und Interoperabilität gewährleistet.

#### Kernapplikation

Die Kernapplikation steht im Zentrum und bildet die fachliche Domäne der Justiz ab. Sie verwaltet die fachlichen Informations- und Metaobjekte und implementiert die Geschäftslogik mit den Regeln und Abläufen der Fachverfahren. Damit ist sie für alle fachlichen Operationen verantwortlich, die zu Änderungen an den Informations- und Metaobjekten führen. Sie stellt die konsistente Abbildung und Verarbeitung der fachlichen Prozesse sicher und grenzt sich damit klar von allgemeinen oder technischen Komponenten ab. Innerhalb der Domäne sind mehrere fachlich klar abgegrenzte Bereiche vorhanden, die jeweils einem Bounded Context im Sinne von DDD entsprechen. Jeder Bounded Context verfügt über eine eigene Sprache, Datenhaltung und Logik und wird technisch durch einen oder mehrere Services umgesetzt. Diese

Services bilden gemeinsam die Fachlogik, können unabhängig entwickelt und betrieben werden und erscheinen nach aussen als eine einheitliche Kernapplikation.

Die Kernapplikation veröffentlicht Domänenereignisse wie „Urteil erstellt“ über den zentralen Eventhub, empfängt Ereignisse aus anderen Komponenten und reagiert darauf durch interne Services, die Arbeitsabläufe automatisch auslösen oder aktualisieren. Alle Ereignisse, sowohl solche zwischen den Bounded Contexts innerhalb der Domäne als auch solche, die domänenübergreifend relevant sind, werden über den zentralen Eventhub veröffentlicht und konsumiert. Dadurch wird eine einheitliche Ereignisinfrastruktur sichergestellt und die Entkopplung der Kontexte unterstützt.

Die Benutzeroberfläche wird im folgenden als Bestandteil der Kernapplikation betrachtet. Sie nutzt deren Schnittstellen für Abfragen und Aktionen und erhält Aktualisierungen über von der Kernapplikation bereitgestellte Mechanismen auf Basis der Ereignisse aus dem Eventhub. Eine direkte Anbindung der Benutzeroberfläche an den Eventhub erfolgt nicht, da die Kernapplikation die Ereignisse verarbeitet und in geeigneter Form an die Oberfläche weitergibt.

### Eventhub

Der Eventhub ist die zentrale Kommunikationsplattform der Architektur. Er stellt sicher, dass alle Komponenten zuverlässig und entkoppelt miteinander kommunizieren können.

Seine Aufgaben sind zweifach:

- Er speichert alle eingehenden Ereignisse dauerhaft im Eventstore, sodass sie jederzeit nachvollzogen, erneut verarbeitet oder für Analysen herangezogen werden können.
- Er verteilt die Ereignisse an alle Komponenten, die diese abonniert haben. Dadurch werden sowohl interne als auch übergreifende Abläufe automatisch ausgelöst, ohne dass direkte Abhängigkeiten zwischen den Systemen entstehen.

Der Eventhub ist eine eigenständige, allgemeingültige Infrastrukturkomponente und gehört nicht zur Kernapplikation, da er selbst keine fachliche Logik der Justiz implementiert. Diese Trennung ermöglicht eine klare Abgrenzung der Verantwortlichkeiten: Die Kernapplikation ist für die fachliche Logik und die Erzeugung der Ereignisse zuständig, während der Eventhub deren Transport, Speicherung und Verteilung übernimmt.

### Best-of-Breed-Komponenten und Adapter

Neben der Kernapplikation und dem Eventhub gibt es weitere spezialisierte Systeme, die nach dem Best-of-Breed-Prinzip ausgewählt werden. Breeds sind spezialisierte Module, die zusätzliche Funktionen bereitstellen, aber nicht Teil der fachlichen Kernlogik sind. Sie ergänzen die Kernapplikation um technische oder querschnittliche Fähigkeiten. Im Folgenden werden diese Systeme vereinfacht als 'Breeds' bezeichnet.

Ein Breed ist ein eigenständiges Modul, das eine klar umrissene Funktion erfüllt, die auch unabhängig von der Kernapplikation existieren könnte. Breeds besitzen in der Regel keine eigene fachliche Verantwortung innerhalb der Justizdomäne, sondern unterstützen oder erweitern die Prozesse der Kernapplikation. Beispiele können ein Dokumentenmanagementsystem, ein Übersetzungstool oder ein Statistikmodul sein.

Viele Breeds liegen als Standardsoftware vor und können nicht direkt an den Eventhub angebunden werden, da ihre Schnittstellen oft proprietär oder nicht ereignisorientiert sind. Hier kommen die Adapter ins Spiel. Jeder Breed verfügt über einen eigenen Adapter, der die Kommunikation ermöglicht.

Der Adapter übersetzt Ereignisse bidirektional zwischen dem Eventhub und dem jeweiligen Modul.

Dadurch können auch Systeme eingebunden werden, die ursprünglich nicht für eine ereignisorientierte Architektur entwickelt wurden.

Adapter bilden eine Abstraktionsschicht, die sicherstellt, dass sowohl Standardsoftware als auch individuell entwickelte Module an der Kommunikation teilnehmen können. Sie stellen die Verbindung von der Kernapplikation zu den Breeds her und machen die Gesamtarchitektur flexibel und offen für unterschiedlichste Systeme.

### High-Level Schaubild zur Zielarchitektur

Das folgende Schaubild verdeutlicht den Aufbau:

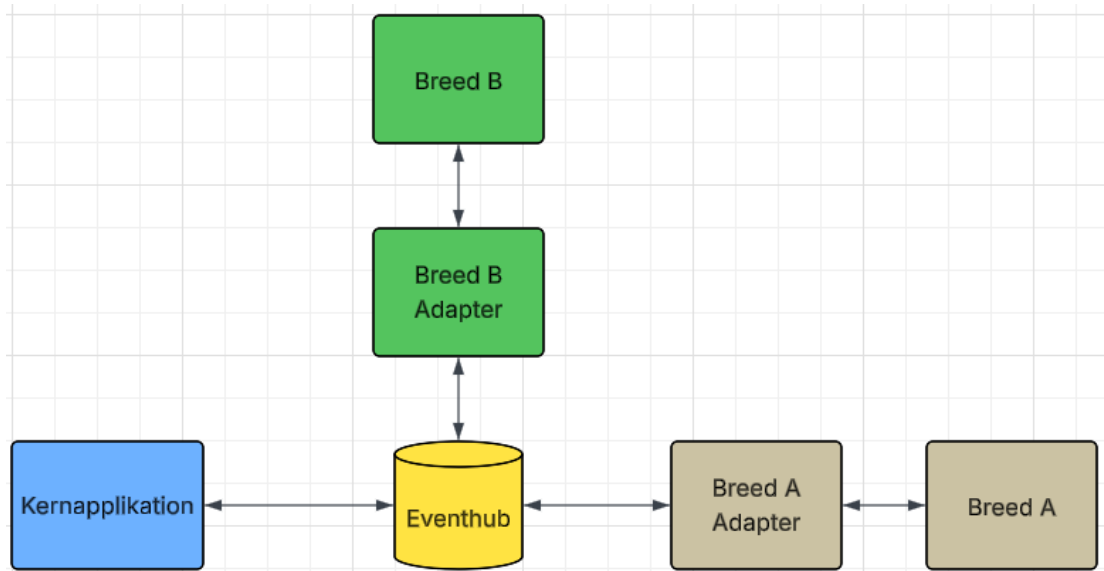


Abbildung 9 High-Level Schaubild zur Zielarchitektur

Die Kernapplikation steht im Zentrum und kommuniziert mit dem Eventhub. Der Eventhub speichert Ereignisse im Eventstore und verteilt sie an die angeschlossenen Adapter. Jeder Breed verfügt über einen eigenen Adapter, der die Ereignisse in ein verständliches Format übersetzt und umgekehrt. Im Beispiel ist zu sehen, wie Breed A über seinen Adapter an den Eventhub angebunden ist und wie Breed B über seinen Adapter mit demselben Eventhub kommuniziert. Dadurch entsteht ein flexibles, modulares und erweiterbares System, das jederzeit um neue Breeds ergänzt werden kann.

#### 5.2.2 Zusammenspiel der Komponenten am Beispielprozess Strafbefehl versenden

Damit die zuvor beschriebenen Architekturprinzipien nicht abstrakt bleiben, sondern in ihrem Zusammenspiel sichtbar werden, wird im Folgenden ein exemplarischer Geschäftsprozess skizziert. Am Beispiel des Urteilsversands lässt sich veranschaulichen, wie Kernapplikation, Eventhub, Adapter und Breeds in der Praxis zusammenwirken.

Der dargestellte Ablauf ist bewusst vereinfacht und erhebt keinen Anspruch auf Vollständigkeit. Er dient ausschliesslich dazu, den Aufbau und die Funktionsweise der Zielarchitektur anschaulich zu machen.

Das folgende Sequenzdiagramm (Abbildung 10 Sequenzdiagramm Strafbefehl versenden) visualisiert die Interaktionen zwischen den beteiligten Komponenten und verdeutlicht den ereignisgesteuerten Ablauf des Prozesses.

Da nicht alle Leserinnen und Leser mit der Notation von Sequenzdiagrammen vertraut sind, wird der gleiche Ablauf anschliessend zusätzlich in tabellarischer Form (Tabelle 2 Tabellarische Beschreibung des Ablaufs) dargestellt. Diese alternative Darstellung ermöglicht es, den Prozess auch ohne technisches Vorwissen nachvollziehen zu können und die Abfolge von Aktionen und Ereignissen leicht nachzuvollziehen.

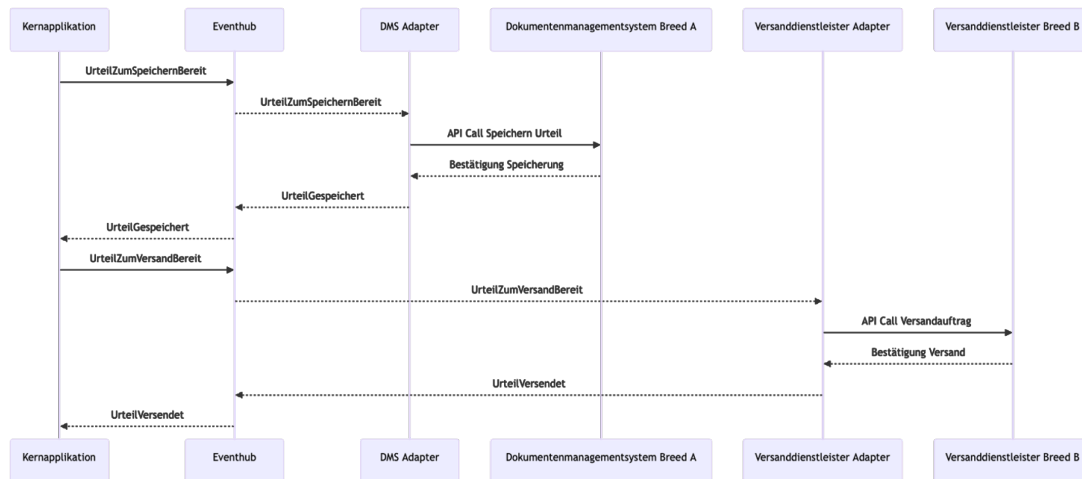


Abbildung 10 Sequenzdiagramm Strafbefehl versenden

| Schritt | Komponente                                       | Aktion  | Event / Ergebnis                |
|---------|--|---|---------------------------------|
| 1       | Kernapplikation                                  | Sachbearbeiter klickt „Urteil versenden“  | Event: UrteilZumSpeichernBereit |
| 2       | DMS-Adapter                                      | Konsumiert Event, transformiert Daten ins DMS-Format                                      | -                               |
| 3       | DMS-Adapter → DMS                                | Übergibt Daten an DMS-API   | -                               |
| 4       | DMS (Breed A)                                    | Speichert das Urteil  | -                               |
| 5       | DMS-Adapter                                      | Nach erfolgreicher Speicherung: publiziert Event  | Event: DokumentGespeichert      |
| 6       | Kernapplikation                                  | Konsumiert Event, validiert Rückmeldung, erzeugt neues Event                              | Event: UrteilZumVersandBereit   |
| 7       | Versand-Adapter                                  | Konsumiert Event, erstellt Versandauftrag   | -                               |
| 8       | Versand-Adapter → Versanddienstleister (Breed B) | Übergibt Versandauftrag an API  | -                               |
| 9       | Versanddienstleister (Breed B)                   | Startet physischen oder digitalen Versand   | -                               |
| 10      | Versand-Adapter                                  | Nach erfolgreichem Versand: publiziert Event  | Event: UrteilVersendet          |
| 11      | Kernapplikation                                  | Konsumiert Event, zeigt Erfolgsmeldung, protokolliert Status, triggert ggf. Folgeprozesse | -                               |

Tabelle 2 Tabellarische Beschreibung des Ablaufs



Hinweis zur tabellarischen Darstellung:

Die tabellarische Darstellung bildet die Abfolge von Aktionen und Ereignissen kompakt und übersichtlich ab. Jeder Schritt ist einer konkreten Komponente zugeordnet, und in der letzten Spalte wird das jeweils resultierende Ereignis aufgeführt. Wo kein neues Ereignis entsteht, ist dies mit einem „–“ gekennzeichnet. Dadurch lassen sich sowohl die Ereigniskette als auch die Rolle der beteiligten Komponenten nachvollziehen.

### Einordnung des Beispiels

Das dargestellte Beispiel macht deutlich, wie die Bausteine der Zielarchitektur in einem realistischen Anwendungsfall zusammenspielen. Es zeigt insbesondere, wie die Kernapplikation Ereignisse erzeugt und darauf reagiert, wie Adapter die Brücke zu bestehenden Systemen schlagen, wie externe Breeds eingebunden werden können, ohne die Architektur zu durchbrechen, und wie Ereignisse entlang des gesamten Prozesses fließen und Transparenz schaffen.

Gerade weil der Ablauf vereinfacht und bewusst unvollständig dargestellt ist, wird der zentrale Gedanke der Architektur deutlich: Die Kernapplikation bleibt entkoppelt von den angeschlossenen Systemen, und alle Interaktionen erfolgen über standardisierte Ereignisse und Adapter.

Dieses Muster lässt sich auf viele andere Geschäftsprozesse übertragen. Dazu zählen beispielsweise die Zustellung eines Urteils, die Erfassung eines Antrags oder die Kommunikation mit einem Statistikmodul. Die Architektur ermöglicht eine modulare, erweiterbare und nachvollziehbare Umsetzung, die unabhängig von den konkret angebundenen Fachsystemen funktioniert.

### 5.2.3 Kernapplikation - Kommunikation und Datenhaltung

Im vorherigen Kapitel wurde die Architektur zunächst abstrakt beschrieben und die einzelnen Komponenten in ihrem Zusammenspiel aufgezeigt. Nun soll der Aufbau der Kernapplikation im Detail betrachtet werden. Zunächst stellt sich die Frage, wie Benutzeroberfläche und Fachservices miteinander kommunizieren. Dieses Kapitel leitet die Entscheidung her und vergleicht zwei unterschiedliche Lösungsvarianten. Ziel ist es zu zeigen, wie Abfragen und Aktionen sicher, nachvollziehbar und performant umgesetzt werden können und wie die Benutzeroberfläche stets aktuell bleibt.

#### Zwei Lösungsvarianten für die Kommunikation zwischen Benutzeroberfläche und Fachservices im Vergleich

Lösungsvariante A: Backend for Frontend (BFF) innerhalb der Kernapplikation

In dieser Variante kommuniziert die Benutzeroberfläche über eine speziell zugeschnittene Schnittstelle, die als BFF bezeichnet wird. Diese Schnittstelle entkoppelt die Oberfläche von der konkreten Implementierung der Services im Hintergrund. Dadurch können Änderungen an fachlichen Services unabhängig vom User Interface erfolgen, ohne dass Anpassungen in der Benutzeroberfläche notwendig werden.

Neben der Entkopplung bietet das BFF die Möglichkeit, Abfragen in einer für die Benutzeroberfläche optimierten Form bereitzustellen. Es fasst Antworten mehrerer Services zusammen, reduziert die Datenmengen auf das Wesentliche und stellt sie so bereit, dass die Benutzeroberfläche schnell reagieren kann. Das BFF übernimmt sowohl Lese- als auch Schreibvorgänge. Lesezugriffe werden über die Fachservices oder deren optimierte Leseansichten verarbeitet. Schreibvorgänge werden über die Schnittstelle an den Eventhub weitergeleitet. Anschliessend können die Ereignisse aus dem Eventhub von den jeweiligen Fachservices verarbeitet werden.

Das BFF fungiert dabei als zentrale Sicherheitsinstanz. Es übernimmt die Authentifizierung und Autorisierung, schützt die internen Services vor direktem Zugriff und ermöglicht eine konsistente Umsetzung von Sicherheitsrichtlinien.

Durch zentrale Logging- und Monitoring-Funktionen wird die Nachvollziehbarkeit erhöht, und die Angriffsfläche reduziert sich auf eine kontrollierte Schnittstelle.

Das Schaubild (Abbildung 11 Schaubild Variante A Backend for Frontend) zeigt diesen Aufbau in einer abstrahierten Form mit drei Modulen innerhalb der Kernapplikation. Es verdeutlicht, wie die Benutzeroberfläche nur eine Schnittstelle ansprechen muss, während die interne Komplexität gekapselt bleibt.

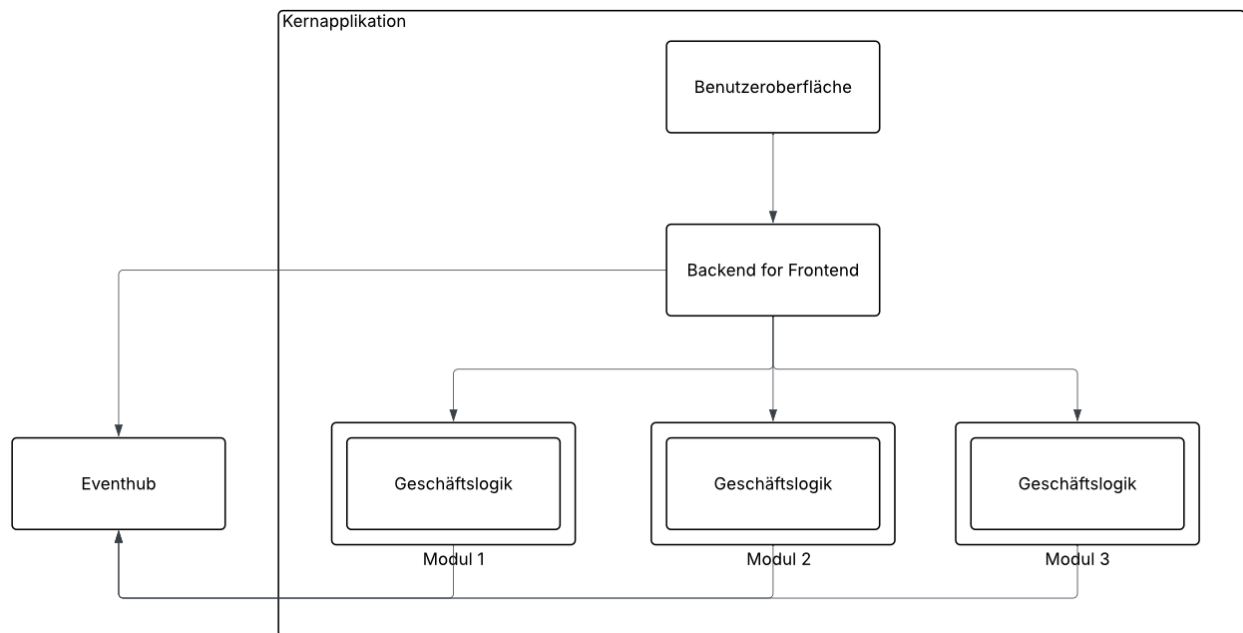


Abbildung 11 Schaubild Variante A Backend for Frontend

#### Hinweis auf mögliche Ausprägung:

Alternativ kann das BFF auch gemeinsam mit einem dedizierten API Gateway eingesetzt werden. In dieser Ausprägung übernimmt das API Gateway zentrale Aufgaben wie Routing, Authentifizierung, Autorisierung, Rate Limiting und Monitoring an der Systemgrenze. Das BFF bleibt weiterhin für die UI-Optimierung und Aggregation der Daten verantwortlich. Durch diese Trennung werden technische und fachliche Verantwortlichkeiten klar abgegrenzt: Das API Gateway sorgt für eine einheitliche, zentral verwaltete Sicherheits- und Governance-Schicht, während das BFF sich auf die spezifischen Anforderungen der Benutzeroberfläche konzentriert. Diese Aufteilung kann insbesondere höheren Anforderungen an Sicherheit, Skalierbarkeit und Wartbarkeit vorteilhaft sein.

#### Lösungsvariante B: Direkter Zugriff der Benutzeroberfläche auf einzelne Services

Die Benutzeroberfläche ruft in dieser Variante mehrere Services direkt auf. Sie muss die Antworten selbst zusammenführen, Fehlerfälle eigenständig behandeln und unterschiedliche Versionen der Services berücksichtigen. Für Aktualisierungen ist sie darauf angewiesen, Ereignisse indirekt nachzuvollziehen oder regelmässig neu zu laden.

Schreibvorgänge werden in diesem Modell durch den Aufruf der jeweiligen Methoden auf den Fachservices ausgeführt. Die Services selbst müssen anschliessend das zugehörige Ereignis an den Eventhub melden, damit es von anderen Modulen weiterverarbeitet werden kann.

In dieser Variante muss jeder Service selbst für Authentifizierung und Autorisierung sorgen. Das führt zu einer verteilten Sicherheitslogik, die schwer zu warten und fehleranfällig ist.

Zudem sind alle Services direkt exponiert, was die Angriffsfläche deutlich erhöht. Einheitliche Sicherheitsrichtlinien lassen sich nur schwer durchsetzen, und die Nachvollziehbarkeit von Zugriffen ist eingeschränkt. Im Schaubild (Abbildung 12 Schaubild Variante B Direkter Zugriff auf Fachservices) ist dies gut zu erkennen: Die Benutzeroberfläche ist mit jedem einzelnen Service direkt verbunden. Dadurch entsteht eine enge Kopplung zwischen UI und fachlichen Services. Die Sicherheitslogik ist verteilt, und jede Änderung in einem Service kann unmittelbare Auswirkungen auf die Oberfläche haben. Dies führt zu höherem Wartungsaufwand, geringerer Flexibilität und einem erhöhten Risiko für Inkonsistenzen im Betrieb.



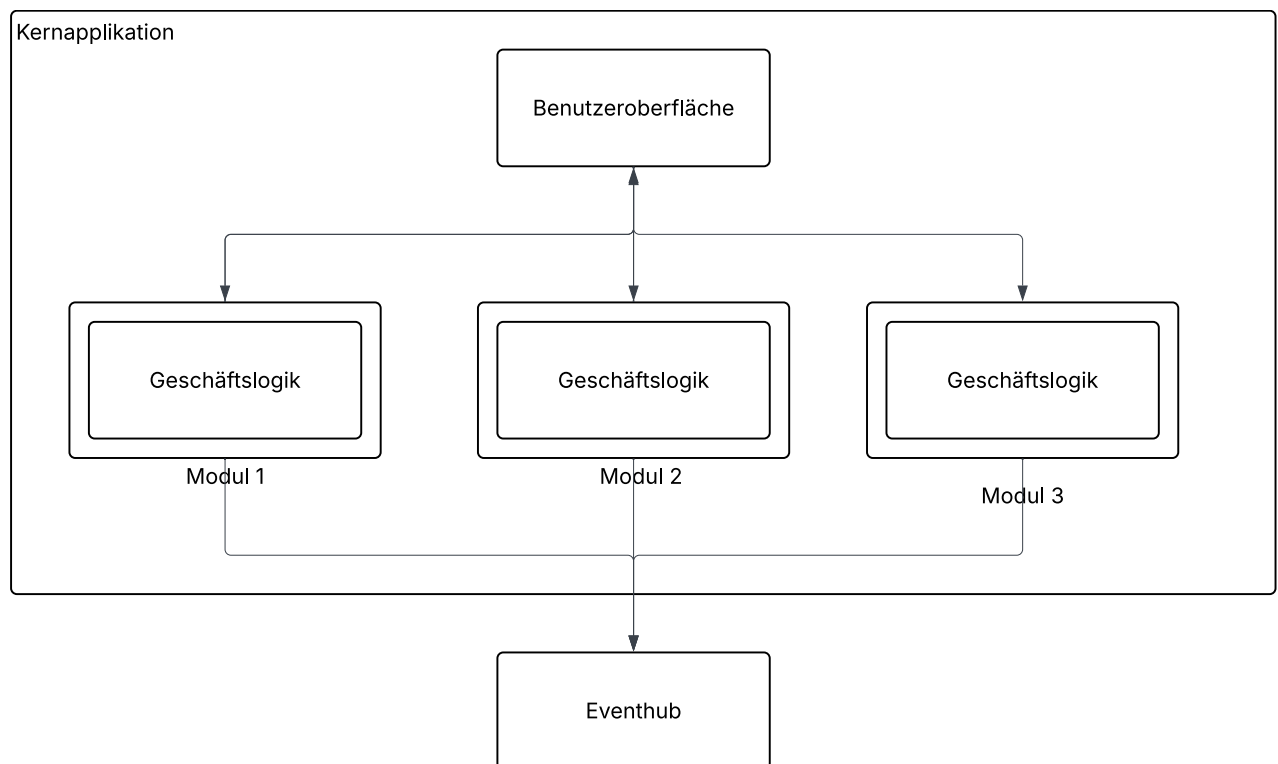


Abbildung 12 Schaubild Variante B Direkter Zugriff auf Fachservices

#### Entscheidung: Variantenvergleich Kommunikation UI und Fachservices

Die folgende Tabelle vergleicht die beiden Varianten anhand der Kriterien Entkopplung und Leistung. Diese Kriterien widerspiegeln die zentralen Anforderungen Performance und Modularität der Nutzer und der Organisationen.

| Kriterium   | Variante A: BFF in der Kernapplikation   | Variante B: Direkter Zugriff der Oberfläche auf Services  |
|-------------|--|---|
| Entkopplung | Die Benutzeroberfläche spricht nur mit einer zentralen Schnittstelle. Interne Änderungen bleiben verborgen | Die Benutzeroberfläche ist direkt mit vielen internen Diensten verbunden. Änderungen wirken sich sofort aus |
| Leistung    | Zusammengefasste Datenansichten, weniger Datenabrufe, schnelle Reaktion durch direkte Benachrichtigungen   | Viele einzelne Datenabrufe, langsamere Reaktion, mehr Aufwand in der Benutzeroberfläche                     |
| Sicherheit  | Zentrale Anmeldung, Rechteprüfung und Protokollierung. Weniger Angriffsflächen                             | Sicherheitsfunktionen sind verteilt, was die Komplexität und das Risiko erhöht                              |

Tabelle 3 Variantenvergleich Kommunikation UI und Fachservices

Aufgrund des Vergleichs zwischen den beiden Lösungsvarianten für die Kommunikation zwischen Benutzeroberfläche und Fachservices wurde Variante A, ein BFF in der Kernapplikation, gewählt. Daten können speziell für die Anforderungen der Oberfläche aufbereitet werden, wodurch die Benutzeroberfläche schneller reagiert und weniger Logik selbst übernehmen muss. Komplexität und Fehlerbehandlung bleiben im Backend gebündelt, die Oberfläche bleibt schlank und stabil. Interne Services können unabhängig weiterentwickelt werden, während die Benutzeroberfläche stabile, UI-optimierte Datenobjekte erhält. Dies verbessert die Nutzererfahrung, erhöht die Wartbarkeit und sichert die langfristige Evolvierbarkeit der Architektur. Ein Nachteil dieser Variante ist die entstehende Kopplung zwischen Benutzeroberfläche und BFF. Änderungen im BFF können direkte Auswirkungen auf die UI haben, was eine enge Abstimmung zwischen

Frontend- und Backend-Teams erfordert. Diese Kopplung wird jedoch bewusst in Kauf genommen, da die Vorteile hinsichtlich Performance, Wartbarkeit und Nutzerfreundlichkeit überwiegen.

Im Idealfall werden sowohl die Benutzeroberfläche als auch das zugehörige BFF vom gleichen Team verantwortet. Dies ermöglicht eine enge Zusammenarbeit, reduziert Kommunikationsaufwand und stellt sicher, dass Anforderungen der UI direkt im BFF berücksichtigt werden können. Dadurch lassen sich Entwicklungszyklen verkürzen und die Qualität der Schnittstellen nachhaltig verbessern.

Die zentrale Sicherheitsarchitektur im BFF erhöht nicht nur die technische Robustheit, sondern auch die Compliance-Fähigkeit der Lösung. Einheitliche Zugriffskontrollen, zentrale Protokollierung und geringere Angriffsflächen sind entscheidende Vorteile für den sicheren Betrieb der Kernapplikation.

Im Gegensatz dazu würde ein direkter Zugriff bedeuten, dass jede Anpassung in den fachlichen Services auch Änderungen in der Benutzeroberfläche erfordert. Dadurch entstünden enge Abhängigkeiten, erhöhter Anpassungsaufwand und eine geringere Flexibilität für zukünftige Entwicklungen.

### Datenhaltung in der Kernapplikation

Nachdem die Frage der Kommunikation zwischen Benutzeroberfläche und Fachservices geklärt ist, betrachten wir als nächstes die Datenhaltung. Neben dem Austausch über Schnittstellen müssen die einzelnen Services auch mit eigenen Daten arbeiten können. Entscheidend ist dabei das Verhältnis zwischen diesen lokalen Beständen und der zentralen Quelle der Wahrheit im Eventstore<sup>2</sup>.

Die Fachservices können jeweils über eine eigene Datenhaltung verfügen. Diese ist jedoch nicht als massgebliche Quelle gedacht, sondern erfüllt eine unterstützende Rolle. Sie stellt optimierte Ablagen für den täglichen Betrieb bereit, etwa für schnelle Abfragen, statistische Auswertungen oder fachlich zugeschnittene Ansichten.

Der Aufbau der Kernapplikation inklusive Eventhub macht dies deutlich:

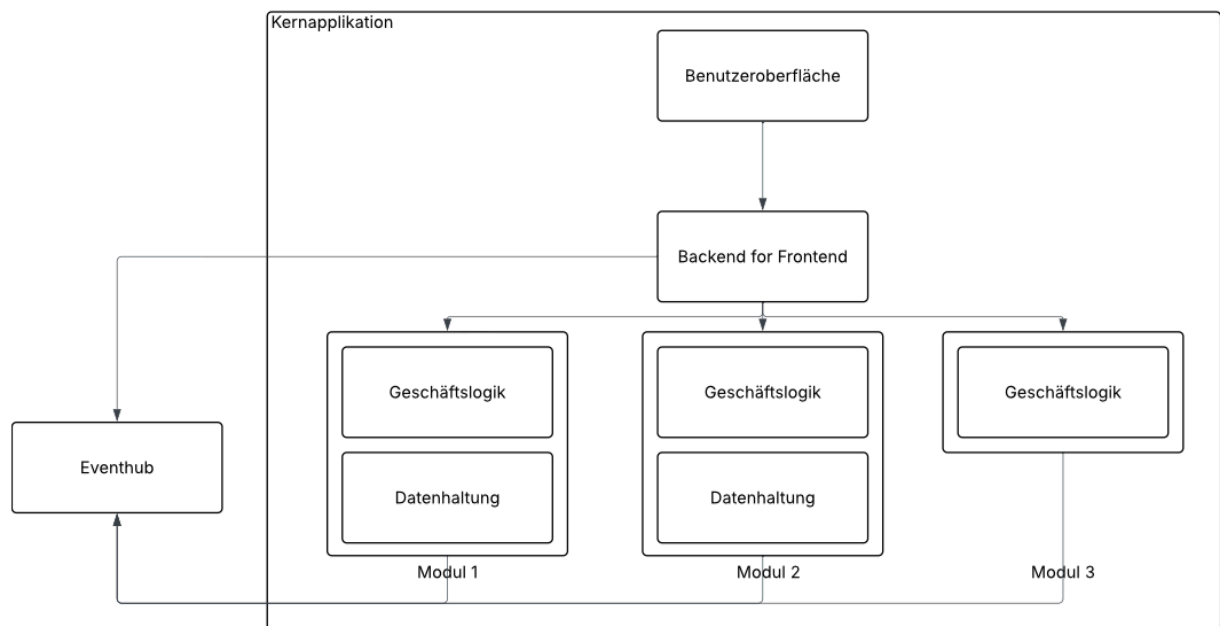


Abbildung 13 Aufbau Kernapplikation inklusive Eventhub

Die Benutzeroberfläche kommuniziert über ein BFF mit verschiedenen Modulen. Jedes Modul enthält die für seinen Aufgabenbereich notwendige Geschäftslogik und kann zusätzlich über eine eigene Datenhaltung verfügen. Ein Service kann also Daten speichern, muss dies aber nicht zwingend tun. Manche Module arbeiten ausschliesslich mit Logik, während andere optimierte Datenbestände anlegen.

Verbindlich bleibt stets der Eventstore. In ihm werden alle relevanten Änderungen an Verfahren als Ereignisse dauerhaft und unveränderlich gespeichert. Aus dieser lückenlosen Historie lässt sich jederzeit der aktuelle Zustand ableiten. Der Eventstore ist damit das rechtlich und fachlich verbindliche Gedächtnis der Architektur.

<sup>2</sup> Der Eventstore kann technisch beispielsweise als SQL-Datenbank realisiert sein. Fachlich handelt es sich um einen Speicher, der sämtliche im System auftretenden Ereignisse persistiert. Ein Dokumentenmanagementsystem (DMS) wäre hingegen ein eigener Breed, der ausschliesslich Aufgaben des Dokumentenmanagements übernimmt.

Die lokale Datenhaltung ergänzt dieses Prinzip, indem sie Lese- und Schreibzugriffe beschleunigt und die Services effizient arbeitsfähig macht. Sollte ein Fachservice seine Daten verlieren oder korrigieren müssen, können diese jederzeit durch das erneute Abspielen der Ereignisse aus dem Eventstore zuverlässig wiederhergestellt werden.

#### 5.2.4 Eventhub: Rolle und Anforderungen

Nachdem nun die Kernapplikation im Detail betrachtet wurde, folgt nun eine nähere Analyse der beiden anderen zentralen Bausteine der Zielarchitektur: Eventhub sowie Breeds mit ihren Adaptern. Während die Kernapplikation das fachliche Herzstück bildet, übernehmen diese Komponenten eine entscheidende Rolle für Kommunikation, Integration und Funktionalität. Um ihre Bedeutung vollständig zu erfassen, werden sie im Folgenden aus fachlicher und technischer Perspektive vertieft beschrieben. Wie bereits im schematischen Aufbau dargestellt, übernimmt der Eventhub die Rolle des zentralen Vermittlers zwischen den Modulen. Er ist das Rückgrat der Architektur, da er sowohl die Kommunikation als auch die dauerhafte Nachvollziehbarkeit fachlicher Abläufe sicherstellt.

##### Fachliche Perspektive

Aus fachlicher Sicht sorgt der Eventhub für Transparenz. Jede Aktivität im System, etwa die Erstellung eines Urteils, die Zustellung eines Bescheids oder die Aktualisierung einer Akte, wird in Form eines Ereignisses über den Eventhub veröffentlicht. Alle interessierten Module können diese Ereignisse abonnieren. So entsteht eine lückenlose Chronologie der Abläufe, die jederzeit ausgewertet und überprüft werden kann.

Man kann sich den Eventhub wie eine Geschäftsstelle in einer Behörde vorstellen: Alle eingehenden Schriftstücke landen dort zuerst und werden anschliessend an die zuständigen Stellen weitergeleitet.

##### Technische Perspektive

Technisch gesehen fungiert der Eventhub als Drehscheibe. Er nimmt Ereignisse entgegen, speichert sie und verteilt sie an alle registrierten Abnehmer. Ein Modul muss dabei weder wissen, von wem das Ereignis kommt, noch wer es später verarbeitet. Diese Entkopplung reduziert Abhängigkeiten und erhöht die Stabilität des Gesamtsystems.

##### Eventstore als Bestandteil

Im abstrakten Schaubild wurde der Eventhub bisher als einheitliche Komponente dargestellt. Tatsächlich besteht er aus zwei Teilen: dem Eventhub selbst, der die Verteilung übernimmt, und dem Eventstore, der für die dauerhafte und unveränderliche Speicherung der Ereignisse sorgt. Diese Zweiteilung ist entscheidend, da sie sicherstellt, dass alle Ereignisse auch Jahre später noch reproduziert werden können. Im folgenden Schaubild (Abbildung 14 Schaubild zur Zielarchitektur - Fokus Eventhub) ist der Eventhub daher mit seinem Eventstore hervorgehoben dargestellt.

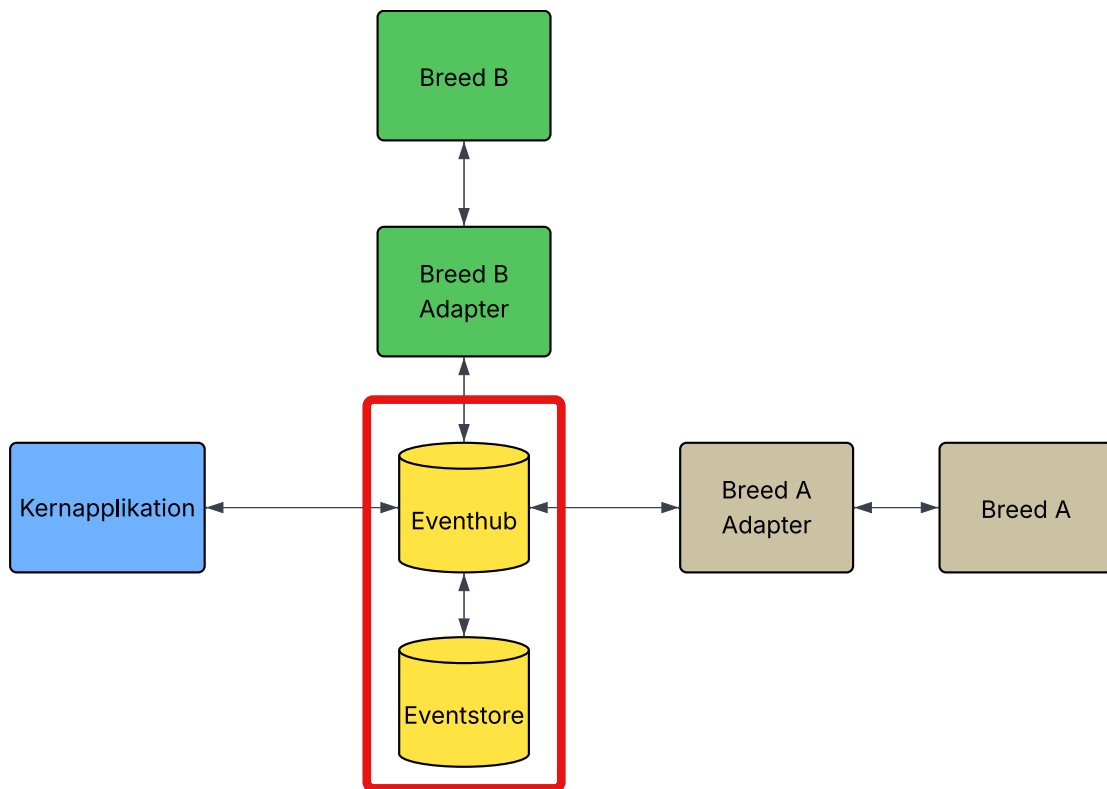


Abbildung 14 Schaubild zur Zielarchitektur - Fokus Eventhub

### Der Eventstore als Gedächtnis der Architektur

Der Eventstore ist mehr als eine Ablage von Nachrichten. Er übernimmt die Funktion des Gedächtnisses der gesamten Architektur. Während der Eventhub für die Verteilung sorgt, stellt der Eventstore sicher, dass jedes Ereignis dauerhaft und unveränderlich gespeichert wird.

Dies hat mehrere Vorteile. Jeder fachliche Schritt von der Einreichung eines Antrags bis zur Zustellung eines Urteils kann auch Jahre später noch nachvollzogen werden. Nicht nur der aktuelle Zustand eines Verfahrens ist sichtbar, sondern auch die gesamte Entstehungsgeschichte. Neue Module können im Nachhinein an bestehende Abläufe angeschlossen werden, indem sie gespeicherte Ereignisse nachträglich konsumieren. Falls ein Modul einen Fehler macht, können die gespeicherten Ereignisse erneut abgespielt werden, um den richtigen Zustand wiederherzustellen.

### Anforderungen an den Eventhub

Damit der Eventhub seine Rolle als zentrales Verteilzentrum in der Applikation zuverlässig wahrnehmen kann, muss das eingesetzte Produkt bestimmte Anforderungen erfüllen.

- **Hohe Belastbarkeit**  
Der Eventhub muss nicht nur mit alltäglichen Vorgängen umgehen können, sondern auch mit aussergewöhnlich vielen gleichzeitigen Ereignissen, wie sie etwa bei Fristenläufen oder Massenverfahren auftreten. Auch unter solchen Bedingungen muss er stabil bleiben und korrekt arbeiten.
- **Dauerhafte Speicherung**  
Alle Ereignisse müssen sicher gespeichert werden, unveränderlich bleiben und mit einem eindeutigen Zeitpunkt versehen sein. So ist auch Jahre später noch nachvollziehbar, wann genau ein bestimmter Schritt im Verfahren stattgefunden hat.
- **Sichere Fehlerbehandlung**  
Falls einzelne Ereignisse fehlerhaft sind oder nicht verarbeitet werden können, darf das Gesamtsystem nicht zum Stillstand kommen. Der Eventhub muss in der Lage sein, solche Ereignisse vorübergehend beiseitezulegen, bis sie geklärt sind.
- **Zuverlässige Zustellung**  
Jedes Ereignis muss sicher bei den zuständigen Modulen ankommen. Sollte es dabei zu doppelten

Zustellungen kommen, darf dies keine falschen Ergebnisse erzeugen. Das System muss so gestaltet sein, dass es mit solchen Situationen korrekt umgehen kann.

- **Transparenz und klare Regeln**  
Es muss jederzeit erkennbar sein, welche Ereignisse im Umlauf sind. Damit die Zusammenarbeit verschiedener Module funktioniert, braucht es klare Regeln dafür, wie Ereignisse benannt und dokumentiert werden. Ausserdem muss es möglich sein, korrigierte Ereignisse gezielt erneut in den Fluss einzuspeisen.

### Produktauswahl

Einfache Nachrichtensysteme wie RabbitMQ oder ActiveMQ erfüllen manche dieser Punkte, stossen jedoch bei den Anforderungen an Speicherung, Fehlerbehandlung und hohe Last an ihre Grenzen. Für eine zukünftige Fachapplikation braucht es deshalb eine Lösung, die sowohl technisch robust ist als auch den rechtlichen Rahmenbedingungen entspricht.

Heutzutage bevorzugen die Behörden OnPremise Lösungen, um Datenhoheit und Kontrolle zu gewährleisten. Gleichzeitig soll die Lösung offen und flexibel genug sein, um auch in anderen Betriebsumgebungen eingesetzt werden zu können.

Aus diesen Gründen bietet sich Apache Kafka als Eventhub an. Kafka ist Open Source und wird weltweit als De-facto Standard für ereignisorientierte Architekturen genutzt. Es vereint Eventhub und Eventstore in einem System, kombiniert hohe Performance mit langlebiger Persistenz, unterstützt die revisionssichere Protokollierung von Ereignissen und ermöglicht durch seine verteilte Architektur eine nahezu unbegrenzte Skalierung. Zudem bringt Kafka Mechanismen für Fehlerbehandlung, Wiederverarbeitung und Dead Letter Queues bereits mit. Da Kafka OnPremise betrieben werden kann, fügt es sich in die Infrastruktur der Justiz ein, ohne Abhängigkeiten von einzelnen Cloud Anbietern zu schaffen. Gleichzeitig bleibt die Option offen, Kafka in Cloud- oder Hybridumgebungen einzusetzen.

Die Wahl von Apache Kafka als Eventhub verbindet die fachlichen Anforderungen mit den organisatorischen Präferenzen. Kafka stellt nicht nur den zentralen Kommunikationsknoten dar, sondern ist zugleich ein verlässlicher Eventstore.

### 5.2.5 Breeds und Adapter im Detail

Moderne Fachapplikationen bestehen nicht nur aus der Kernapplikation, die die fachlichen Informations- und Metaobjekte verwaltet, sondern nutzen eine Vielzahl spezialisierter Systeme, die bestimmte Aufgaben besonders effizient erfüllen. In dieser Studie werden diese Systeme basierend auf dem Best-of-Breed Ansatz nur Breeds genannt. Sie ergänzen die Fachapplikation um unterstützende, oft technische oder querschnittliche Fähigkeiten.

### Begriff und Abgrenzung

Der Begriff Breed bezeichnet in dieser Zielarchitektur ein funktionsspezifisches Ergänzungsmodul ausserhalb der fachlichen Kernlogik. Ein Breed stellt eine klar umrissene Fähigkeit bereit, kann unabhängig von der Fachapplikation betrieben werden und wird über einen Adapter in die ereignisorientierte Kommunikation integriert.

Es wird dabei zwischen Fähigkeit und Instanz unterschieden. Die Fähigkeit beschreibt die Funktionalität des Breed. Die Instanz bezeichnet die konkrete Anwendung oder Produkt für das Erreichen dieser Funktionalität. Der Begriff Best-of-Breed bezieht sich dabei ausschliesslich auf die Auswahl der jeweils besten Instanz für eine Fähigkeit.

Die Kernkomponenten Kernapplikation, Eventhub und Eventstore werden nicht als Breeds betrachtet. Sie tragen die fachliche Logik bzw. die zentrale Kommunikations- und Persistenzfunktion und nehmen daher eine eigene, übergeordnete Rolle in der Architektur ein.

Ein Modul wird als Breed klassifiziert, wenn es:

- eine abgegrenzte Fähigkeit liefert, die auch unabhängig von der Fachapplikation bestehen kann und
- auf Instanzebene austauschbar bleibt

Die Entscheidung, ob eine Funktion in der Kernapplikation oder als Breed umgesetzt wird, hängt von ihrer fachlichen Verantwortung und technischen Kopplung ab.

- **Kernapplikation**  
Enthält die fachliche Logik der Domäne, verwaltet Zustände mit rechtlicher oder fachlicher Bedeutung und erzeugt die Domänenereignisse. Funktionen gehören hierher, wenn sie unmittelbar Teil der fachlichen Abläufe sind oder deren Konsistenz beeinflussen.
- **Breed**  
Übernimmt ergänzende, technische oder querschnittliche Aufgaben. Breeds verarbeiten oder erweitern Ereignisse der Kernapplikation, beeinflussen jedoch nicht deren fachliche Zustände.

#### Fachliche Perspektive

Aus fachlicher Sicht sind Breeds spezialisierte Werkzeuge, die Aufgaben übernehmen, welche nicht in der Kernapplikation selbst abgebildet werden sollen. Ein Dokumentenmanagementsystem dient beispielsweise der sicheren Speicherung und Verwaltung von Schriftstücken. Ein Übersetzungsdienst gewährleistet, dass Dokumente in der richtigen Sprache vorliegen. Ein Versanddienstleister übernimmt die Zustellung von Urteilen und Bescheiden, während ein Statistikmodul Auswertungen über Verfahrensabläufe ermöglicht. Alle diese Systeme tragen entscheidend dazu bei, dass die Fachapplikation den Bedürfnissen von Anwendern und Organisationen gerecht wird. Ohne Breeds würde die Kernapplikation entweder unnötig komplex oder wichtige Funktionen wären gar nicht verfügbar.

#### Technische Perspektive

Technisch ergibt sich jedoch ein Problem. Viele mögliche Instanzen von Breeds sind Standardprodukte, die unabhängig von der Zielarchitektur entstanden sind und eigene Integrationslogiken mitbringen. Sie kommunizieren über proprietäre Schnittstellen oder erwarten eine synchrone Verarbeitung, die nicht zur ereignisgesteuerten Arbeitsweise passt.

An dieser Stelle kommen die Adapter ins Spiel. Ein Adapter ist ein Integrationsmodul, das zwischen Kernapplikation und Breed vermittelt. Er nimmt Ereignisse aus dem Eventhub entgegen, übersetzt sie in ein Format, das der Breed versteht, und überträgt sie an dessen Schnittstelle. Umgekehrt nimmt er die Antworten oder Rückmeldungen des Breeds entgegen und publiziert sie wieder als Ereignis in den Eventhub. Der Adapter ist damit nicht nur ein Übersetzer, sondern auch ein Stabilitätsanker. Er kapselt die Besonderheiten des jeweiligen Systems und stellt sicher, dass die übrige Architektur davon unberührt bleibt. Dies macht die Gesamtarchitektur flexibel und offen für unterschiedlichste Systeme. Für die Kernapplikation ist es nicht relevant, ob das Dokumentenmanagementsystem seine Daten über eine REST-Schnittstelle, eine SOAP-Schnittstelle oder eine proprietäre API entgegennimmt. Die Kernapplikation interagiert ausschliesslich über standardisierte Ereignisse mit dem Adapter.

Um die Vielfalt und Spezialisierung der Breeds in der Zielarchitektur zu veranschaulichen, zeigt die nachfolgende Übersicht (Tabelle 4 Übersicht Breeds) der Breeds mit ihren jeweiligen Fähigkeiten. Sie verdeutlicht, wie breit das Spektrum unterstützender Funktionen ausserhalb der Kernapplikation ist.

| Name            | Fähigkeit  |
|-----------------|--|
| DMS             | Dokumentenmanagementsystem zur zentralen Ablage, Verwaltung und Versionierung von Dokumenten     |
| Archiv          | Langzeitaufbewahrung von Dokumenten und Daten gemäss gesetzlichen Vorgaben.                      |
| Signatur        | Digitale Signatur zur rechtsverbindlichen Unterzeichnung von Dokumenten.                         |
| Trägerwandel    | Verwaltung und Nachverfolgung von Trägerwechseln innerhalb eines Systems oder einer Organisation |
| Fallbuchhaltung | Finanzielle Abwicklung und Buchführung auf Fallbasis, zum Beispiel für Sozialleistungen          |
| Statistik       | Erhebung, Auswertung und Visualisierung von Daten zur Entscheidungsunterstützung                 |

|                                       |  |
|---------------------------------------|--|
| newVostra                             | Verwaltung von Personen- und Falldaten im Straf- und Massnahmenvollzug   |
| IAM                                   | Identity and Access Management zur Steuerung von Benutzerrechten und -zugängen.  |
| Outputmanagement                      | Verteilung und Nachverfolgung ausgehender Informationen (Papier)   |
| Justitia.Swiss                        | Plattform für den elektronischen Rechtsverkehr in der Schweiz.   |
| Mail                                  | E-Mail-Kommunikation innerhalb und ausserhalb der Organisation   |
| PDF                                   | Erstellung, Anzeige und Bearbeitung von PDF-Dokumenten.  |
| Aufgabenplanung                       | Verwaltung und Zuweisung von Aufgaben zur Unterstützung der Projekt- und Teamarbeit.   |
| Fristenmanagement                     | Überwachung und Einhaltung von Fristen und Terminen (gesetzlicher Hintergrund)   |
| Pendenzenverwaltung                   | Überwachung und Einhaltung von Fristen und Terminen (kein gesetzlicher Hintergrund)  |
| Zellbewirtschaftung                   | Verwaltung von Zellen und Insassen im Strafvollzug   |
| Video-Streaming                       | Übertragung von Video-Inhalten in Echtzeit, zum Beispiel für Anhörungen oder Schulungen.   |
| Publikation                           | Veröffentlichung von Informationen, zum Beispiel im Amtsblatt oder auf Webseiten.  |
| Rechtsrecherche                       | Recherche in juristischen Datenbanken und Gesetzestexten.  |
| ZEMIS-Schnittstelle                   | Zentrales Migrationsinformationssystem der Schweiz zur Verwaltung von Ausländer- und Asylgesuchen.   |
| Personenregister                      | Sammlung strukturierter Personendaten zur eindeutigen Identifikation und Dokumentation rechtlicher sowie tatsächlicher Verhältnisse            |
| GERES Schnittstelle                   | Schnittstelle zum Gemeinsamen Register der Einwohnerdienste für den Datenaustausch zwischen Behörden   |
| Adressstämme                          | Verwaltung und Pflege von Adressdaten zur einheitlichen Nutzung in verschiedenen Anwendungen   |
| Validatoren-Service                   | Prüfung und Validierung elektronischer Signaturen und Zertifikate  |
| Dolmetsch-Verzeichnis                 | Verzeichnis von Dolmetscherinnen mit Bewertungsfunktion zur Qualitätssicherung   |
| Asservaten-Verwaltung                 | Verwaltung und Nachverfolgung von sichergestellten Gegenständen im Strafverfahren  |
| Medikamenten-Verwaltung               | Dokumentation und Verwaltung von Medikamentenausgabe und -beständen, zum Beispiel in Justizvollzugsanstalten                                   |
| Notification-Service                  | Versand von allgemeinen Benachrichtigungen an Benutzer oder Systeme.   |
| Notification-Service (Behörden)       | Umsetzung gesetzlicher Mitteilungspflichten gegenüber Behörden   |
| ISJV                                  | Informationssystem Justizvollzug. Vereinfacht Platzierungen und stellt Daten für Statistiken und operative Prozesse im Schweizer Justizvollzug |
| Ordnungsbussen Schnittstelle generell | Schnittstelle zur Verarbeitung und Verwaltung von Ordnungsbussen.  |



|                                 |   |
|---------------------------------|---|
| Anonymisierung von Texten       | Automatisierte Entfernung oder Maskierung personenbezogener Daten in Texten   |
| Chatbot                         | KI-gestützter Assistent zur Beantwortung von Anfragen und Unterstützung bei Routineaufgaben                                       |
| Generierung von Textvorschlägen | Automatische Erstellung von Formulierungshilfen, zum Beispiel für Schreiben, Berichte oder Stellungnahmen                         |
| Statistik Vorhersage-AI         | KI-gestützte Prognose von Entwicklungen auf Basis statistischer Daten, zum Beispiel Auslastung oder Rückfallrisiken               |
| Klassifikation von Nachrichten  | Automatische Einordnung und Klassifizierung eingehender Mitteilungen  |
| Video-Analyse                   | KI-gestützte Auswertung von Videomaterial, zum Beispiel zur Verhaltensanalyse oder zur Erkennung sicherheitsrelevanter Ereignisse |

Tabelle 4 Übersicht Breeds

Die grosse Vielfalt an Breeds bringt jedoch auch technische und organisatorische Herausforderungen mit sich. Unterschiedliche Technologien, Schnittstellen und Entwicklungsstände erfordern eine sorgfältige Integration, um die Stabilität und Einheitlichkeit der Gesamtarchitektur zu gewährleisten.

#### Herausforderungen der Integration

Die Einbindung von Breeds stellt hohe Anforderungen an Architektur, Betrieb und Governance. Proprietäre Schnittstellen müssen berücksichtigt, bestehende Legacy-Systeme weiterhin unterstützt und neue Komponenten nahtlos integriert werden. Dabei ist sicherzustellen, dass alle Adapter nach denselben Prinzipien arbeiten und einheitliche Schnittstellenkonventionen einhalten.

Adapter sind damit nicht nur technische Hilfsmittel, sondern zentrale Bausteine für Stabilität und Erweiterbarkeit. Sie schützen die Kernapplikation vor Änderungen in den angeschlossenen Systemen, ermöglichen den kontrollierten Austausch einzelner Breeds und sichern die Einhaltung der Architekturprinzipien über Systemgrenzen hinweg. Nur durch eine konsistente Governance und klare Verantwortlichkeiten kann gewährleistet werden, dass die Gesamtarchitektur auch bei wachsender Anzahl von Breeds flexibel, erweiterbar und betriebssicher bleibt.

#### Fazit

Breeds und Adapter bilden zusammen die Brücke zwischen der modularen Architektur und den vorhandenen oder neu hinzukommenden Speziallösungen. Sie ermöglichen es, dass bestehende Systeme weiter genutzt werden können, ohne die Architekturprinzipien zu verletzen. Gleichzeitig schaffen sie die Grundlage für Innovation, da jederzeit neue Breeds angebunden werden können, solange ein passender Adapter bereitgestellt wird.

#### 5.2.6 Justizaktenapplikation – JAA

Die Justizakten-Applikation (JAA) ist als zentrale digitale Lösung für die Schweizer Justiz konzipiert und befindet sich aktuell in der Einführung. Ziel ist es, künftig die vollständige elektronische Verwaltung und Bearbeitung von Akten zu ermöglichen und damit die papierbasierte Aktenführung abzulösen. Die JAA wird momentan schrittweise bei mehreren Kantonen und bei Bundesbehörden eingeführt. Sie stellt ein zentrales Element bei diesen Organisationen dar. Weitere Behörden prüfen den Einsatz im Moment ebenso. Somit muss eine zukünftige Fachapplikation die JAA bei Bedarf integrieren können.

Die JAA bündelt dabei mehrere Fähigkeiten und würde somit mehrere Breeds im Sinne der Konzeptstudie darstellen. Zu den zentralen Fähigkeiten der JAA zählen:

- Elektronische Aktenführung und Dokumentenmanagement
- Dokumentenbearbeitung und Annotation
- Aufgaben- und Fristenmanagement
- Intuitive Akteneinsicht und Suche
- Zugriffs- und Rechteverwaltung



Die Nutzung der JAA ist sehr gut möglich, da sie eine Schnittstelle bietet. Wie andere Breeds wird sie über einen Adapter an die jeweilige Fachapplikation angebunden und kann so flexibel in unterschiedliche Systemlandschaften integriert werden.

Um weiterhin flexibel zu bleiben, kann für jede einzelne Fähigkeit der JAA ein eigener Adapter entwickelt werden. Dadurch ist es möglich, bei Bedarf spezielle Fähigkeiten später zu ergänzen oder einzelne Funktionen durch spezialisierte Produkte zu ersetzen. Die Integration muss dabei nicht komplett neugestaltet werden. So bleibt die Architektur offen für Innovationen und Anpassungen. Dieses Vorgehen entspricht dem Best-of-Breed-Prinzip und stellt sicher, dass die Fachapplikation auch langfristig an neue Anforderungen angepasst werden kann.

### 5.2.7 Abschluss des Kapitels Zielarchitektur Fachapplikation

Mit der Zielarchitektur liegt nun ein konsistenter Bauplan vor, der die zentralen Prinzipien Modularisierung, Ereignisorientierung, Event Sourcing und Standardisierung in eine konkrete Struktur überführt. Wie in Abbildung 15 Exemplarischer Aufbau der Zielarchitektur der Fachapplikation dargestellt, zeigt die Kernapplikation gemeinsam mit dem Eventhub, dem Eventstore sowie den Breeds und Adaptern, wie technische und fachliche Anforderungen zusammenwirken und sich gegenseitig ergänzen.

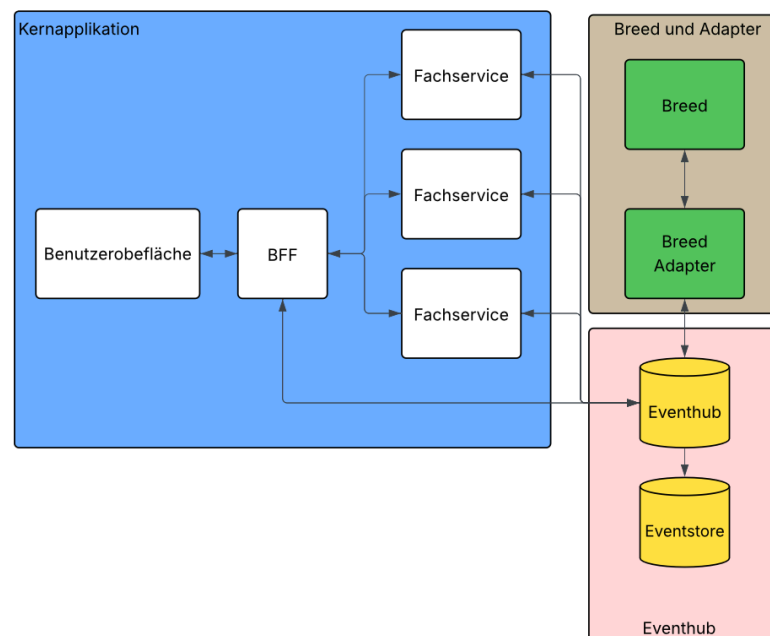


Abbildung 15 Exemplarischer Aufbau der Zielarchitektur der Fachapplikation

Das exemplarische Beispiel zum Urteilsversand hat verdeutlicht, wie diese Bausteine im praktischen Ablauf zusammenspielen und wie durch Ereignisse Transparenz und Nachvollziehbarkeit entstehen. Gleichzeitig wurde sichtbar, dass die Architektur nicht starr ist, sondern durch die Adapter flexibel auf neue Systeme und Anforderungen reagieren kann.

Die festgelegten Architekturprinzipien wie Modularisierung, DDD sowie Governance und Standardisierung wurden von Vertretern verschiedener Institutionen im öffentlichen Umfeld bestätigt und als sehr geeigneter Ansatz für die Entwicklung zukunftsgerichteter Fachapplikationen bewertet. Damit stützen die Rückmeldungen die Tragfähigkeit und Relevanz der gewählten Richtung.

Damit bildet die Zielarchitektur nicht nur die Grundlage für Fachapplikationen, sondern auch für die langfristige Weiterentwicklung der Justiz-informatik. Sie schafft Stabilität und Rechtssicherheit und eröffnet zugleich den notwendigen Spielraum für Innovationen und zukünftige Erweiterungen.

Ein Bauplan allein genügt jedoch nicht. Entscheidend ist, wie dieser im Betrieb umgesetzt und weiterentwickelt wird. Damit die in der Zielarchitektur beschriebenen Prinzipien ihre Wirkung entfalten können, braucht es ein passendes Betriebsmodell. Es sorgt dafür, dass Modularisierung, Ereignisorientierung und Standardisierung nicht nur theoretische Konzepte bleiben, sondern im Alltag zuverlässig getragen werden.

## 6 Betriebsmodell in einer modularen, ereignisgesteuerten Justizarchitektur

### 6.1 Ausgangslage und Herausforderungen

Die Einführung moderner Softwarearchitekturen, wie sie in der Zielarchitektur beschrieben wurde, markiert einen Paradigmenwechsel im IT-Betrieb der Justiz. Insbesondere der Einsatz von Microservices und EDA eröffnet erhebliche Vorteile hinsichtlich Wartbarkeit, Zukunftsfähigkeit und Performance. Gleichzeitig stellen sie jedoch höhere Anforderungen an Betriebsorganisationen, insbesondere in Bezug auf:

- Automatisierung und Überwachbarkeit (Monitoring, Logging, Alerting)
- Resilienz und Fehlertoleranz (Recovery-Strategien, Event-Replay)
- Technische Entkopplung (lose Kopplung, asynchrone Kommunikation)
- Sicherheitsarchitektur (Mandantentrennung, Zugriffskontrolle, Auditierbarkeit)

Diese Anforderungen sind in der Justizlandschaft auf sehr heterogene Betriebsrealitäten zu treffen: Während klassische IT-Kompetenzen (zum Beispiel Windows-Server, VMs, relationale Datenbanken) weit verbreitet sind, fehlt es vielerorts an Erfahrung mit containerisierten Umgebungen, orchestrierten Deployments (zum Beispiel Kubernetes) und ereignisgesteuerten Systemen.

### 6.2 Notwendigkeit differenzierter Betriebsmodelle

Die Einführung moderner, auf Microservices und ereignisgesteuerter Architektur basierender Softwaresysteme in der Justiz erfordert ein Umdenken in der Betriebsorganisation. Diese Architekturen bringen nicht nur technische Neuerungen mit sich, sondern stellen auch erhöhte Anforderungen an Überwachung, Automatisierung, Sicherheit und organisatorische Steuerung.

Dabei zeigt sich: Ein einheitliches Betriebsmodell ist weder realistisch noch zielführend. Die Justizlandschaft ist geprägt von sehr unterschiedlichen technischen Ausstattungen, strategischen Zielsetzungen und Kooperationsbereitschaften. Während einige Organisationen über etablierte IT-Abteilungen mit hoher technischer Reife verfügen, stehen andere erst am Anfang der Transformation hin zu containerisierten, modularen Systemen.

Um tragfähige Lösungen zu ermöglichen, ist daher ein mehrstufiges Betriebsmodell erforderlich, das sowohl technische Reifegrade als auch organisatorische Kapazitäten berücksichtigt.

Folgende organisatorische Modelle bieten sich an:

1. Inhouse-Betrieb durch die einzelne Organisation
  - für maximale Autonomie und Datenhoheit
2. Kooperationsmodell mehrerer Organisationen
  - für geteilte Verantwortung und gemeinsame Plattformnutzung
3. Betrieb durch einen externen Dienstleister für mehrere Mandanten
  - für professionellen, skalierbaren Betrieb mit klaren SLAs

Diese Modelle sind nicht als Konkurrenz, sondern als komplementäre Optionen zu verstehen. Sie ermöglichen es, den Betrieb modularer Justizsoftware flexibel an die jeweiligen Rahmenbedingungen anzupassen und dabei sowohl Sicherheit als auch Effizienz und Innovationsfähigkeit zu gewährleisten.

### 6.2.1 Modell 1: Inhouse-Modell



Abbildung 16 Inhouse-Modell - Betriebsmodell

#### Beschreibung

Die Organisation betreibt die Fachapplikation inklusive Infrastruktur, Applikationen, Monitoring, Backup und Sicherheitsmassnahmen eigenständig. Sie trägt die volle Verantwortung für den Betrieb und die Anpassung der Systeme.

#### Beispiel

- Kantonale Staatsanwaltschaft mit ihrer eigenen IT-Abteilung

#### Chancen

- Maximale Datenhoheit: Alle Daten verbleiben innerhalb der Organisation
- Hohe Flexibilität: Anpassungen an spezifische Anforderungen sind jederzeit möglich
- Unabhängigkeit: Keine Abhängigkeit von anderen Mandanten

#### Risiken

- Komplexität im Betrieb: Event-driven Microservices erfordern tiefes technisches Verständnis für Software
- Sicherheitsverantwortung: Die Organisation muss selbst für grosse Teile der Sicherheit der Applikation sorgen
- Ressourcen: Die Betriebsorganisation kann personell und technisch von den Anforderungen überfordert sein

#### Vorteile

- Volle Kontrolle über Konfigurationen und Betriebsprozesse
- Möglichkeit zur individuellen Optimierung und Integration
- Geeignet für sensible Fachverfahren mit besonderen Anforderungen

#### Nachteile

- Höhere Betriebskosten durch fehlende Synergien
- Erhöhter Schulungs- und Aufbauaufwand für Personal

## 6.2.2 Modell 2: Kooperationsmodell mehrerer Organisationen



Abbildung 17 Kooperationsmodell mehrerer Organisationen - Betriebsmodell

### Beschreibung

Eine Organisation betreibt die Anwendung für mehrere Mandanten. Der technische Betrieb erfolgt zentral, während die Mandantentrennung durch technische Massnahmen wie separate Datenbankschemata und Eventstreams sichergestellt wird.

### Beispiel

- Kantonale Gerichtsbehörde mit eigener IT übernimmt den Betrieb der Fachapplikation für die eigene Organisation wie auch für die Steuerrekurskommission

### Chancen

- Effizienzsteigerung: Gemeinsame Nutzung reduziert Kosten und Aufwand
- Innovationsförderung: Neue Funktionen stehen allen Beteiligten zur Verfügung

### Risiken

- Abhängigkeit von zentraler Betriebsorganisation: Reaktionsfähigkeit bei Störungen kann eingeschränkt sein
- Governance-Komplexität: Rollen und Verantwortlichkeiten müssen klar geregelt sein
- Vertrauensfragen: Mandantentrennung und Datenschutz müssen transparent und auditierbar sein

### Vorteile

- Skaleneffekte durch gebündelte Ressourcen
- Professionalisierter Betrieb durch zentrale Expertise
- Schnellere Weiterentwicklung durch koordinierte Prozesse

### Nachteile

- Eingeschränkte Individualisierungsmöglichkeiten
- Erhöhter Abstimmungsbedarf zwischen Organisationen
- Potenzielle Konflikte bei Priorisierung gemeinsamer Entwicklungen

### 6.2.3 Modell 3: Betrieb mehrerer Mandanten bei einem externen Dienstleister



Abbildung 18 Betrieb mehrerer Mandanten bei einem externen Dienstleister - Betriebsmodell

#### Beschreibung

Ein externer, spezialisierter IT-Dienstleister übernimmt den vollständigen technischen Betrieb der Softwareplattform für mehrere Justizorganisationen (Mandanten). Der Dienstleister stellt Infrastruktur, Betriebsprozesse, Monitoring, Sicherheit und Support zentral bereit. Diese Art von Betrieb ist auch als SaaS-Betrieb bekannt

#### Beispiel

- Unternehmungen oder interkantonale Körperschaft betreibt die Fachapplikation zentral für mehrere nutzende kantonalen Justizbehörden

#### Chancen

- Professionalisierung des Betriebs: Der Dienstleister bringt spezialisierte Expertise in modernen Betriebsmodellen (zum Beispiel Container-Orchestrierung, Event-Streaming, DevSecOps) ein
- Schnelle Skalierbarkeit: Neue Mandanten können zügig integriert werden, ohne dass lokale Ressourcen aufgebaut werden müssen
- Konzentration auf Fachlichkeit: Die Justizorganisationen können sich auf ihre fachlichen Aufgaben konzentrieren, während der technische Betrieb ausgelagert ist
- Förderung von Innovation: Der Dienstleister kann eventuell moderne Technologien schneller einführen und betreiben, zum Beispiel KI-gestützte Analysefunktionen oder KI-Modelle
- Verfügbarkeit von SLAs: Verbindliche Service Level Agreements schaffen Transparenz und Verlässlichkeit im Betrieb

#### Risiken

- Abhängigkeit vom Dienstleister: Die Organisationen sind auf die Leistungsfähigkeit und Verfügbarkeit des externen Partners angewiesen
- Datenschutz und Souveränität: Die Auslagerung sensibler Daten erfordert strenge vertragliche, technische und organisatorische Schutzmassnahmen
- Komplexe Vertragsgestaltung: Die Anforderungen an Sicherheit, Verfügbarkeit, Auditierbarkeit und Governance müssen detailliert geregelt werden
- Verlust von Betriebs-Know-how: Langfristig kann internes technisches Wissen verloren gehen

#### Vorteile

- Reduktion des internen Betriebsaufwands
- Zugang zu hochqualifizierten Betriebsressourcen und modernen Technologien
- Einheitliche Betriebsstandards über alle Mandanten hinweg
- Möglichkeit zur Kostenbündelung und -senkung durch Skaleneffekte

## Nachteile

- Eingeschränkte Einflussmöglichkeiten auf Betriebsprozesse
- Höherer Abstimmungsbedarf bei Änderungen oder Störungen
- Erhöhte Anforderungen an Datenschutz- und Sicherheitskontrollen
- Potenzielle Konflikte bei Priorisierung von Weiterentwicklungen im Multi-Mandantenbetrieb

## 6.2.4 Technische Betriebsgrundlagen

Im nächsten Abschnitt wird zunächst erläutert, welche Betriebstools für ereignisgesteuerte Architekturen besonders relevant sind und wie sie die Betriebsfähigkeit moderner Justizanwendungen gezielt unterstützen können. Anschliessend wird aufgezeigt, welche Betriebsumgebungen wie zum Beispiel Public Cloud, Private Cloud oder der Betrieb im eigenen Rechenzentrum für den Einsatz dieser Tools geeignet sind und welchen Einfluss die Wahl der Umgebung auf Flexibilität, Skalierbarkeit und Wartbarkeit der Lösung hat. Während das gewählte Betriebsmodell den organisatorischen Rahmen vorgibt, bestimmt die technische Umsetzung im Detail, wie flexibel, skalierbar und wartbar eine Lösung im Alltag betrieben werden kann. Dabei spielt insbesondere die Frage eine Rolle, wie die einzelnen Bestandteile der Zielarchitektur (Fachanwendung, Eventhub, modulare Adapter) technisch orchestriert und betrieben werden. Für den stabilen und effizienten Betrieb ereignisgesteuerter Architekturen ist der Einsatz geeigneter Betriebstools entscheidend. Insbesondere bei modular aufgebauten Systemen mit klar getrennten Komponenten wie Fachanwendungen, Eventhub und technischen Adaptern stellt sich die Frage, wie diese Bausteine orchestriert, skaliert und überwacht werden können. In der Praxis haben sich dabei insbesondere zwei Ansätze bzw. Plattformen als besonders geeignet erwiesen, um die Betriebsfähigkeit solcher Architekturen sicherzustellen.

### Containerbasierte Bündelung der Microservice in Komponenten

Eine Möglichkeit besteht darin, alle wesentlichen Bausteine wie Kernapplikation, Eventhub und Adapter in eigenen Containern zu betreiben. Ein Container ist ein kleines, abgeschlossenes Softwarepaket, das eine Anwendung samt aller benötigten Bestandteile wie Bibliotheken und Konfiguration enthält. Man kann sich Container wie genormte Transportboxen vorstellen, die überall gleich aussehen und die Anwendung überall gleichlaufen lassen.

Werden mehrere Container gemeinsam gestartet, entsteht eine gebündelte Komponente aus mehreren Bausteinen, die zusammen eine vollständige Anwendung bilden (siehe Abbildung 19 Aufbau der Anwendung als gebündelte Komponente mit einzelnen Containern).

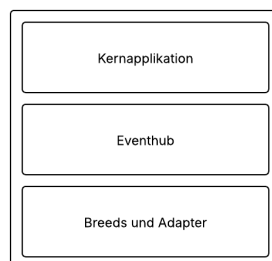


Abbildung 19 Aufbau der Anwendung als gebündelte Komponente mit einzelnen Containern

### Werkzeuge zur Umsetzung

Diese Container können mit Werkzeugen wie Docker, Podman oder containerd betrieben werden. Alle drei haben die gleiche Aufgabe, unterscheiden sich aber in der Handhabung:

- Docker ist weit verbreitet, einfach zu bedienen und daher besonders für den Einstieg und kleinere Setups geeignet.
- Podman erlaubt den Betrieb ohne zentrale Hintergrunddienste und erhöht so die Sicherheit. Das ist im behördlichen Kontext wichtig, wo Angriffsflächen reduziert werden sollen.
- Containerd ist sehr schlank und bildet im Hintergrund die Basis vieler moderner Plattformen. Es eignet sich besonders für automatisierte Szenarien.



### Vorteile

Der grösste Vorteil dieses Ansatzes liegt in seiner Einfachheit. Container lassen sich leicht starten und beenden, sind schnell bereitzustellen und sehr gut nachvollziehbar. Zudem sorgt die Standardisierung der Container dafür, dass Anwendungen in unterschiedlichen Umgebungen zuverlässig laufen, unabhängig davon, ob sie auf einem Server im eigenen Rechenzentrum oder in einer Cloud betrieben werden. Auch die Trennung der einzelnen Container erhöht die Stabilität, da Fehler in einem Baustein nicht automatisch das gesamte System betreffen. Damit bietet dieser Ansatz einen vergleichsweise geringen administrativen Aufwand bei gleichzeitig hoher Transparenz.

### Nachteile

Die Nachteile zeigen sich vor allem bei komplexeren Szenarien. Die Möglichkeit, einzelne Teile unabhängig voneinander hoch- oder herunterzufahren, ist nur eingeschränkt gegeben, was die Skalierbarkeit limitiert. Ebenso lassen sich neue Versionen einer Anwendung nicht ohne Weiteres während des laufenden Betriebs einspielen. Rolling Updates, bei denen eine Anwendung Schritt für Schritt erneuert wird, während die alte Version weiterhin aktiv bleibt, sind nur bedingt möglich. Für die Nutzer bedeutet dies, dass Aktualisierungen in dieser Variante spürbare Unterbrechungen verursachen können.

Die Werkzeuge zur Umsetzung verfügen zwar über grundlegende Mechanismen, mit denen abgestürzte Container automatisch neu gestartet werden können. Eine umfassende Selbstheilung wie sie Plattformen mit integriertem Orchestrierungsansatz bieten, ist hier jedoch nicht vorhanden. Dazu zählen Funktionen wie Lastverteilung, der automatische Start von Ersatzinstanzen oder die Verlagerung auf andere Server. Mit wachsender Anzahl von Containern steigt zudem die Komplexität der Verwaltung, da jedes System einzeln überwacht und gepflegt werden muss. Damit ist die containerbasierte Bündelung vor allem für überschaubare Szenarien geeignet, stösst aber bei grossen, verteilten Anwendungen schnell an organisatorische und technische Grenzen.

### Betrieb auf einer Containerplattform

Eine Containerplattform verwaltet die einzelnen Bestandteile einer Anwendung als eigenständige Microservices. Fachlogik, Eventverarbeitung und technische Adapter laufen jeweils in eigenen Containern. Die Plattform übernimmt die Orchestrierung, also die automatische Steuerung und Koordination dieser Container. Sie platziert Container auf geeigneten Servern, startet sie bei Bedarf neu, skaliert sie passend zur Last und hält den Gesamtzustand der Anwendung stabil.

Man kann sich die Plattform wie einen Dirigenten vorstellen. Viele einzelne Musiker spielen gemeinsam ein Stück, ohne dass jeder allein den Überblick behalten muss. Die Plattform verteilt Lasten automatisch, startet Ersatzinstanzen, wenn ein Dienst ausfällt, und stabilisiert den Betrieb über mehrere Server hinweg.

### Werkzeuge zur Umsetzung

- Kubernetes ist die verbreitete Grundlage. Es stellt Planung, Platzierung, Skalierung, Selbstheilung, Service Abstraktion und Konfiguration bereit
- Red Hat OpenShift basiert auf Kubernetes und ergänzt integrierte Sicherheitsfunktionen, Build und Deploy Werkzeuge, ein durchgängiges Rechte Modell und eine einheitliche Oberfläche für Entwicklung und Betrieb

### Vorteile

Der grösste Vorteil dieses Ansatzes liegt in der Flexibilität und Betriebssicherheit. Da jeder Microservice einzeln verwaltet wird, können Fachapplikation, Eventhub oder Adapter unabhängig voneinander skaliert und aktualisiert werden. Neue Versionen lassen sich mit Rolling Updates einspielen, ohne dass es zu Unterbrechungen für die Nutzer kommt.

Ein weiterer Vorteil ist die Selbstheilung. Die Plattform überwacht alle laufenden Container automatisch und startet fehlgeschlagene Dienste neu. Sie kann Lasten über mehrere Server hinweg verteilen und sicherstellen, dass wichtige Funktionen jederzeit verfügbar bleiben. Dadurch entsteht eine deutlich höhere Stabilität im Vergleich zur containerbasierten Bündelung.

Auch die Erweiterbarkeit ist ein Pluspunkt. Neue Module oder Funktionen lassen sich hinzufügen, ohne die bestehenden Systeme anpassen zu müssen. Das ist besonders in grossen Organisationen von Vorteil, in denen unterschiedliche Teams parallel an verschiedenen Teilen der Anwendung arbeiten.

#### Nachteile

Die Vorteile gehen mit höheren Anforderungen einher. Der Betrieb auf einer Containerplattform erfordert eine komplexere Infrastruktur. Einrichtung, Überwachung, Protokollierung und Wiederherstellung müssen umfassender geplant werden und erfordern spezialisierte Betriebstools. Auch die Sicherheitskonzepte müssen detaillierter ausgestaltet sein, etwa durch Rollen- und Rechtemodelle oder verschlüsselte Kommunikation zwischen den Microservices.

Ein weiterer Nachteil ist der höhere organisatorische Aufwand. Der Betrieb einer Plattform wie Kubernetes verlangt spezielles Know-how im Umgang mit containerisierten Umgebungen, Orchestrierung und Automatisierung. In vielen Justizorganisationen sind diese Kompetenzen heute noch nicht flächendeckend vorhanden, sodass zunächst in Schulung und Aufbau von Fachwissen investiert werden muss.

### 6.3 Verknüpfung von Betriebsmodell und Betriebstools: Ein Auswahlprozess

#### 6.3.1 Ausgangspunkt: Zwei Dimensionen

Die Wahl des organisatorischen Betriebsmodells (Inhouse, Kooperation, externer Dienstleister) kann nicht isoliert von den technischen Betriebsgrundlagen (Container-Bündelung, Containerplattform) getroffen werden. Ein Inhouse-Betrieb mit geringen Ressourcen stellt andere Anforderungen an die Technik als ein professioneller Multi-Mandanten-Betrieb durch einen externen Dienstleister. Ebenso bringt der Betrieb auf einer Containerplattform organisatorische Konsequenzen mit sich, etwa den Bedarf an spezialisiertem Know-how und abgestimmten Governance-Strukturen.

Es entsteht damit eine gegenseitige Abhängigkeit:

- Das organisatorische Modell legt fest, welche Freiheitsgrade, Steuerungs- und Sicherheitsansprüche bestehen.
- Die technische Basis bestimmt, wie skalierbar, resilient und automatisierbar des Betriebs tatsächlich umgesetzt werden kann.

Nur wenn beide Dimensionen zusammen gedacht werden, ergibt sich ein tragfähiges, realistisch betreibbares Modell.

#### 6.3.2 Kriterien und Entscheidungsprozess zur Auswahl des passenden Betriebsmodells

Die Wahl eines geeigneten Betriebsmodells für eine modulare, ereignisgesteuerte Justizarchitektur ist ein strategischer Schritt mit weitreichenden Auswirkungen auf Betriebssicherheit, Innovationsfähigkeit und Ressourceneinsatz. Um dieser Entscheidung die nötige Tiefe und Systematik zu verleihen, empfiehlt sich ein mehrstufiger Auswahlprozess, der sowohl technische als auch organisatorische Faktoren berücksichtigt.

#### Analyse des technischen Reifegrades

- Besteht Erfahrung mit containerisierten Umgebungen?
- Gibt es etablierte Prozesse für Monitoring, Logging und Automatisierung?
- Ist Know-how im Umgang mit Event-Streaming vorhanden?

Konsequenz:

Ist der technische Reifegrad gering, können containerisierte Anwendungen zwar genutzt werden, ein vollwertiger Containerplattformbetrieb ist jedoch kaum realistisch. In diesem Fall bietet sich entweder ein internes Modell mit einfacher Container-Bündelung an oder die Auslagerung an einen Dienstleister, der eine Plattform bereitstellt und betreibt.

Liegt hingegen ein hoher Reifegrad vor, also vorhandene Erfahrung mit Containern, etabliertes Betriebs-Know-how und Automatisierungsprozessen, ist ein Betrieb einer Containerplattform im eigenen Haus

möglich. Alternativ kann in diesem Fall auch eine Kooperation mit Partnern zur gemeinsamen Nutzung einer Plattform sinnvoll sein.

#### Bewertung der organisatorischen Kapazitäten

- Stehen qualifizierte Betriebsmitarbeitende bereit?
- Gibt es klare Verantwortlichkeiten für Sicherheit, Recovery und Incident Management?
- Besteht Bereitschaft zur Zusammenarbeit mit anderen Organisationen?

##### Konsequenz:

Verfügt die Organisation über stark ausgeprägte personelle und organisatorische Ressourcen, kann eine Containerplattform im Eigenbetrieb realisiert werden.

Sind die Ressourcen vorhanden, aber in Umfang oder Spezialisierung begrenzt, eignet sich eine kooperative Lösung, bei der eine gemeinsame Plattform mit anderen betrieben wird.

Stehen nur wenige interne Ressourcen zur Verfügung, ist ein eigenständiger Betrieb nicht tragfähig, so dass in der Regel ein Dienstleistermodell mit Containerplattform erforderlich ist.

#### Berücksichtigung fachlicher und rechtlicher Anforderungen

- Gibt es gesetzliche Vorgaben zur Datenhoheit oder Auditierbarkeit?
- Besteht hoher Bedarf an individueller Anpassung?

##### Konsequenz:

Sind strenge gesetzliche Vorgaben zur Datenhoheit oder umfangreiche Anforderungen an individuelle Anpassungen gegeben, ist ein Inhouse-Modell erforderlich. Technisch bedeutet dies in der Regel ein Plattformbetrieb mit klarer und strikter Trennung der Systeme, um Sicherheit und Nachvollziehbarkeit zu gewährleisten.

Bestehen hingegen vorwiegend standardisierte Verfahren ohne hohen Individualisierungsbedarf, kann eine Kooperation oder auch die Nutzung eines Dienstleistermodells erfolgen. Technisch wird dabei eine mandantenfähige Plattform betrieben, die getrennte, aber standardisierte Umgebungen für verschiedene Organisationen bereitstellt.

#### Strategische Zielsetzung und Innovationsbereitschaft

- Soll IT-Betrieb als Kernkompetenz verstanden werden?
- Besteht Interesse an schnellen Innovationszyklen?
- Wird Standardisierung über mehrere Organisationen angestrebt?

##### Konsequenz:

Steht die Eigenständigkeit im Vordergrund und soll der IT-Betrieb als Kernkompetenz verstanden werden, ist ein Inhouse-Modell mit einer Containerplattform die passende Wahl. Dies ermöglicht maximale Kontrolle und Eigenverantwortung.

Wenn hingegen Innovationsgeschwindigkeit, Skalierbarkeit und die Möglichkeit zur Standardisierung über mehrere Organisationen im Fokus stehen, ist ein kooperatives Modell oder die Zusammenarbeit mit einem Dienstleister naheliegend. In beiden Fällen bildet ein Plattformbetrieb die technische Grundlage, um Skaleneffekte und schnelle Weiterentwicklung zu ermöglichen.

#### Ergebnisoffene Anwendung des Prozesses

Der Entscheidungsprozess führt nicht zu einem starren Ergebnis, sondern unterstützt dabei, die passende Kombination aus organisatorischem Modell und technischer Basis für die jeweilige Situation zu finden. Organisationen mit hohen Anforderungen an Datenhoheit und Eigenständigkeit neigen eher zu einem Inhouse-Modell, technisch entweder mit einer einfacheren Container-Bündelung oder mit einer professionellen Containerplattform. Kooperationen sind besonders dann sinnvoll, wenn Ressourcen geteilt und Skaleneffekte genutzt werden sollen, was fast immer auf den Einsatz einer Containerplattform hinausläuft. Externe Dienstleister sind schliesslich prädestiniert für standardisierte Verfahren mit hoher Skalierbarkeit und können durch ihre Expertise in der Regel moderne Plattformlösungen effizient betreiben.

Auch hybride Szenarien wären möglich: So kann eine Organisation sensible Verfahren weiterhin inhouse betreiben, während standardisierte Anwendungen bei einem Dienstleister laufen. Entscheidend ist stets der bewusste Abgleich von organisatorischen Möglichkeiten, fachlichen Anforderungen und technischen Potenzialen.

Die Analyse der möglichen Betriebsmodelle und der damit verbundenen Herausforderungen im vorherigen Kapitel hat deutlich gemacht, dass die Einführung einer modularen, ereignisgesteuerten Architektur nicht nur technische, sondern auch organisatorische und personelle Anforderungen an die betreibende Organisation stellt. Insbesondere die Integration zahlreicher spezialisierter Komponenten (Best-of-Breed) sowie die Sicherstellung von Interoperabilität, Wartbarkeit und Weiterentwicklung erfordern ein hohes Mass an Koordination und spezifischem Know-how.

Gerade für Organisationen, die über begrenzte Ressourcen oder wenig Erfahrung im Betrieb komplexer IT-Landschaften verfügen, stellt sich daher die Frage, wie diese Herausforderungen effizient und nachhaltig bewältigt werden können. An dieser Stelle bietet sich das Generalunternehmermodell als mögliche Lösung an: Es ermöglicht, die Gesamtverantwortung für Planung, Integration und Betrieb in die Hände eines zentralen Partners zu legen und so die Komplexität für die Behörde deutlich zu reduzieren.

Im folgenden Kapitel wird daher das Generalunternehmermodell als Option für eine effiziente Entwicklung und Umsetzung moderner IT-Lösungen in der Schweiz näher beleuchtet und konkrete Anforderungen an einen GU abgeleitet.

## 7 Generalunternehmermodell – Steuerung, Integration und Effizienz in der Projektumsetzung

Ein Generalunternehmer (GU) ist ein Unternehmen, das als zentraler Vertragspartner die Gesamtverantwortung für die Planung, Entwicklung und Umsetzung eines komplexen IT-Projekts übernimmt. Im Unterschied zu einer Vielzahl einzelner Dienstleister koordiniert der Generalunternehmer sämtliche beteiligten Subunternehmen und Spezialisten und bleibt für die Auftraggeberseite während aller Projektphasen der Hauptansprechpartner. Die Behörde muss dadurch nicht selbst die Steuerung und Integration verschiedener Anbieter übernehmen, sondern kann sich auf einen zentralen Partner verlassen. Der Leistungsumfang eines Generalunternehmers umfasst das Anforderungsmanagement und die Beratung, das Architektur- und Lösungsdesign, die Entwicklung und Integration der Softwaremodule, die Qualitätssicherung, das Projektmanagement, die Kommunikation mit dem Auftraggeber sowie die Betriebsüberführung und den Support.

Gerade im Kontext einer Best-of-Breed-Anwendung, bei der verschiedene spezialisierte Softwarekomponenten zu einer Gesamtlösung integriert werden, bietet der Einsatz eines Generalunternehmers besondere Vorteile. Die Komplexität der Integration unterschiedlicher Systeme, die Abstimmung von Schnittstellen und die Sicherstellung der Interoperabilität werden zentral gesteuert. Dadurch kann gewährleistet werden, dass alle Komponenten nahtlos zusammenarbeiten und die angestrebte Flexibilität und Zukunftsfähigkeit der Lösung tatsächlich erreicht werden. Der GU übernimmt die Koordination der beteiligten Anbieter, sorgt für die Einhaltung von Standards und Schnittstellen und trägt die Verantwortung dafür, dass die Gesamtlösung den fachlichen und technischen Anforderungen entspricht.

Ein weiterer wichtiger Aspekt ist die Unterstützung beim Know-how-Aufbau auf Behördenseite. Der GU kann durch gezielte Schulungen und begleitende Unterstützung dazu beitragen, dass die Behörde die notwendigen Kompetenzen entwickelt, um die Lösung eigenständig betreiben und weiterentwickeln zu können. Solange das erforderliche Wissen noch nicht ausreichend vorhanden ist, kann er operativ unterstützen und als Ansprechpartner zur Verfügung stehen. Damit dieser Know-how-Transfer tatsächlich gelingt, sollte er explizit als Ziel und Leistung im Vertrag festgehalten werden. Es ist wichtig, dass die Behörde aktiv einfordert, dass Schulungen und Dokumentationen auf hohem Niveau bereitgestellt werden, um eine nachhaltige Unabhängigkeit von externen Dienstleistern zu fördern.

Die Verantwortung für Termine, Qualität und Kosten liegt gebündelt beim GU, was die Projektdurchführung effizienter und die Risikoverteilung klarer macht. Wichtig ist jedoch, dass die Vergabe an einen Generalunternehmer nicht zu einer neuen Abhängigkeit führt. Die Vertragsgestaltung sollte daher Offenheit für modulare Erweiterungen, klare Schnittstellen und die Möglichkeit zum späteren Anbieterwechsel sicherstellen. Gleichzeitig bleibt es essenziell, dass die Behörde eigenes Know-how aufbaut, um die Leistungen des Generalunternehmers kompetent steuern und bewerten zu können.

### 7.1 Anforderungen an potenzielle Generalunternehmer

Nachdem zuerst die Rolle des Generalunternehmers für die effiziente Entwicklung und Integration komplexer IT-Lösungen herausgearbeitet wurde, stellt sich nun die Frage, welche konkreten Anforderungen an einen solchen Partner gestellt werden müssen. Nur wenn der GU über das notwendige Leistungsspektrum

und die passenden Kompetenzen verfügt, kann er die vielfältigen Herausforderungen eines Best-of-Breed-Ansatzes erfolgreich bewältigen und die Ziele der öffentlichen Hand nachhaltig unterstützen.

Dafür ist es sinnvoll, die Anforderungen in zwei Kategorien zu unterteilen: technische und nicht-technische Anforderungen. Zunächst werden die technischen Anforderungen beleuchtet, die sich auf Architektur, Integration, Sicherheit und Betrieb beziehen. Im Anschluss daran werden die nicht-technischen Anforderungen betrachtet, die insbesondere organisatorische, kommunikative und strategische Aspekte abdecken.

### 7.1.1 Technische Anforderungen an den Generalunternehmer

Um in einem solchen Kontext als Generalunternehmer agieren zu können, müssen eine Reihe von technischen Anforderungen erfüllt werden. Diese Anforderungen stellen sicher, dass der GU über die notwendige Expertise verfügt, um die in dieser Studie entwickelten Konzepte und Architekturprinzipien in die Praxis zu übertragen. Gleichzeitig gewährleisten sie, dass die zu entwickelnde Fachapplikation die geforderte Qualität und Güte erreicht und langfristig betreibbar sowie erweiterbar bleibt.

Technische Anforderungen im Überblick:

- Erfahrung mit Best-of-Breed-Integrationen:  
Kompetenz in der Integration und dem Betrieb von Lösungen, die aus mehreren spezialisierten Komponenten bestehen
- Modularität und offene Schnittstellen  
Erfahrung mit der Entwicklung von modularen Applikationen unter Nutzung von klar definierten Schnittstellen (zum Beispiel OpenAPI, AsyncAPI)
- Ereignisgesteuerte-Architektur  
Erfahrung in der Konzeption, Implementierung und dem Betrieb von ereignisgesteuerten Architekturen. Dazu gehört insbesondere die Fähigkeit, modulare Komponenten über Events zu integrieren, Eventhub und Eventstore zu nutzen sowie asynchrone Kommunikationsmuster (zum Beispiel mit Apache Kafka oder vergleichbaren Technologien) umzusetzen
- Sicherheitsstandards  
Erfahrung und Nutzung aktueller Sicherheitsstandards (zum Beispiel TLS, SSO, rollenbasierte Zugriffskontrolle)
- Qualitätssicherung und Testmanagement  
Erfahrung und Nachweis der Nutzung strukturierter QS-Prozesse, automatisierter Tests und regelmäßiger Reviews zur Sicherstellung der langfristigen Systemstabilität und -qualität
- Dokumentation  
Alle Architekturen, Schnittstellen, Konfigurationen und Betriebsprozesse müssen umfassend und verständlich dokumentiert
- Langfristige Projektarbeiten  
Erfahrung in der Durchführung von langfristigen Projektarbeiten inkl. Konzept zur Sicherstellung vor Wissensverlust
- Entwicklung von Applikationen mit Betrieb in unterschiedlichen Umgebungen  
Fähigkeit eine Lösung sowohl On-Premise als auch in Private-Cloud-Umgebungen zu betreiben und zu warten

Diese Anforderungen stellen einen Rahmen dar, welche von einem GU technisch erfüllt werden sollten. Diese ergeben sich aus standardmässigen Anforderungen aus komplexen IT-Projekten wie aber auch Anforderungen welche sich aus dem Umfeld der Justiz wie auch den Konzepten der Studie ergeben

### 7.1.2 Nicht-technische Anforderungen an einen Generalunternehmer

Die bisherigen Kapitel der Studie haben deutlich gemacht, dass die erfolgreiche Umsetzung einer modularen, zukunftsfähigen Fachapplikation für die Justiz weit mehr als technisches Know-how erfordert. Gerade im komplexen und föderal geprägten Umfeld der Schweizer Justiz sind organisatorische, kommunikative und strategische Kompetenzen des GU mindestens ebenso entscheidend wie die technische Expertise. Die nachfolgenden Anforderungen leiten sich direkt aus den Herausforderungen, Zielen und Handlungsfeldern der Studie ab.

Nicht-technische Anforderungen:

- **Projektmanagement und Steuerungskompetenz**  
Nachweisbare Erfahrung in der Leitung komplexer, langlaufender IT-Projekte mit vielen Beteiligten. Fähigkeit zur Etablierung klarer Projektstrukturen, aktivem Risikomanagement
- **Kommunikation und Stakeholder-Management**  
Sicherstellung einer transparenten, regelmässigen und adressatengerechten Kommunikation mit allen relevanten Stakeholdern. Übernahme der Rolle als zentraler Ansprechpartner und Moderator
- **Know-How-Transfer und Schulung**  
Erfahrungen und Nachweise von Systematischem und fortlaufendem Wissenstransfer in Kundenprojekten zum Beispiel durch praxisnahe Schulungen und Workshops
- **Innovationsbereitschaft**  
Bereitschaft und Fähigkeit, innovative Ansätze wie Open-Source einzubringen
- **Vor-Ort-Präsenz**  
Fähigkeit zur Vor-Ort-Betreuung, insbesondere in der Konzeption und Rollout-Phase
- **Kenntnisse der Justiz**  
Kenntnis der Verwaltungskultur Justiz sowie der föderalen Besonderheiten der Schweiz
- **Skalierende Umsetzung**  
Notwendige Grösse und Fähigkeit mehrere Kantone und Behörden gleichzeitig zu betreuen

Die konsequente Berücksichtigung dieser Anforderungen stellt sicher, dass die Behörde nicht nur eine leistungsfähige Fachapplikation erhält, sondern auch die notwendige Unabhängigkeit, Flexibilität und Steuerungsfähigkeit für die Zukunft gewinnt.

## 7.2 Auslegeordnung Generalunternehmer

Um zu bewerten, ob der Markt entsprechende Möglichkeiten bietet und wie potenzielle Anbieter auf die in dieser Studie entwickelten Konzepte, Ideen und Anforderungen reagieren, wurden semi-strukturierte Interviews mit mehreren Unternehmen durchgeführt. Neben etablierten Softwaredienstleistern wie ti&m und Zühlke wurden auch innovative Anbieter aus dem Justizbereich, wie beispielsweise CaseDoc, in die Gespräche einbezogen. Während ti&m AG und Zühlke AG bereits über umfangreiche Projekterfahrung und Referenzen im öffentlichen Sektor der Schweiz oder im Justizbereich verfügen, steht CaseDoc aktuell noch am Beginn der Markterschliessung im Schweizer Justizmarkt.

Durch diese Auswahl an Interviewpartnern konnten sowohl die Sichtweisen und Kompetenzen erfahrener Dienstleister als auch die Potenziale und Herausforderungen neuer Marktteilnehmer in die Bewertung einfließen. Die nachfolgende Auslegeordnung fasst die wichtigsten Erkenntnisse aus diesen Interviews zusammen.

Die mit den Unternehmen geführten Interviews zeigen ein hohes strategisches Interesse an der Bereitstellung von Fachapplikationen für den Schweizer Justizmarkt. Auch wenn einige Unternehmen heute noch nicht im Justizmarkt aktiv waren, besteht ein reges Interesse, in diesen Markt einzutreten. Alle befragten Anbieter haben bestätigt, dass sie bereit sind, die Rolle des Generalunternehmers zu übernehmen und die Gesamtverantwortung für Planung, Integration und Betrieb aus einer Hand zu liefern. Dabei wurde jedoch deutlich, dass die Strukturen für den langfristigen Support und Betrieb, wie etwa ein dediziertes Betriebsteam, nicht bei allen Anbietern bereits vorhanden sind. Ein Anbieter sieht sich aktuell eher als Dienstleister im Projektgeschäft und weniger als Produkthersteller, was darauf hindeutet, dass der Aufbau entsprechender Supportstrukturen im Markt noch nicht überall abgeschlossen ist. Diese Erkenntnis sollte in die Anforderungen an einen Generalunternehmer zurückfliessen, insbesondere im Hinblick auf die Sicherstellung eines nachhaltigen Betriebs und Supports.



Ein weiterer wichtiger Punkt ist das spezifische Wissen über den Schweizer Justizmarkt und die fachlichen Anforderungen der Justiz. Nicht alle Anbieter verfügen bereits über explizites Justizwissen oder tiefgehende Kenntnisse der Schweizer Justizlandschaft. In solchen Fällen wäre es sinnvoll, dieses Know-how durch eine zweite Partei, beispielsweise einen spezialisierten Partner oder Berater, einzubringen. Alternativ könnten auch langfristige Betriebs- und Supportstrukturen so gestaltet werden, dass das notwendige Domänenwissen kontinuierlich aufgebaut und gesichert wird.

Einige Anbieter verfügen bereits über einsetzbare Digitalisierungsplattformen oder Module, die unmittelbar in Projekten verwendet werden können. Die Interviewpartner bestätigten die Architekturideen der Studie und betonten, dass modulare Applikationen sich im Markt von Softwaresystemen durchgesetzt haben und bei allen die Standardmethode für den Aufbau von Applikationen darstellen. Alle Anbieter konnten zudem Erfahrungen im Vorgehen mit ereignisgesteuerter Architektur vorweisen und sehen diese Ansätze als sehr valide und zukunftsfähig an.

Ein weiteres zentrales Thema ist der Umgang mit Open-Source-Ansätzen. Die Offenheit gegenüber Open Source ist grundsätzlich vorhanden und die Nutzung wird von den Anbietern als sehr wichtig erachtet. Es besteht ein breiter Konsens, dass Open-Source-Komponenten insbesondere im Kontext von Standardisierung und Interoperabilität eine wichtige Rolle spielen können.

Alle befragten Dienstleister konnten zudem Konzepte für die technische Sicherstellung der Qualität einer Software darlegen. Bezüglich Referenzen verfügen mehrere Anbieter über einschlägige Erfahrungen in vergleichbaren Kontexten. Dazu zählen insbesondere Projekte im Bereich ereignisgesteuerter Architekturen, die Entwicklung und der Betrieb von Multi-Tenant-Plattformen sowie erfolgreiche Umsetzungen im Justiz- und Behördenumfeld. Diese Referenzen werden von den Anbietern als Beleg für ihre technische und organisatorische Kompetenz hervorgehoben.

Die Einschätzung von Chancen und Risiken fällt differenziert aus. Als zentrale Chancen werden die Erschliessung neuer Use Cases, die Möglichkeit zur Standardisierung und die Erzielung von Kosteneffekten durch modulare und wiederverwendbare Komponenten genannt. Gleichzeitig werden aber auch Risiken gesehen, insbesondere im Hinblick auf den Integrationsaufwand heterogener Systeme, die technische Komplexität moderner Architekturen sowie die nachhaltige Pflege von Open-Source-Communities.

Die Interviews bestätigen, dass der Markt grundsätzlich bereit ist, innovative und zukunftsfähige Lösungen für den Justizbereich zu entwickeln. Die in der Studie definierten Anforderungen an Architektur, Modularität, ereignisgesteuerte Architektur, Qualitätssicherung und Offenheit für Open Source werden von den Anbietern geteilt und als valide eingeschätzt. Allerdings zeigt sich, dass sowohl die langfristige Sicherstellung von Betrieb und Support als auch das spezifische Justizwissen nicht bei allen Anbietern bereits vollständig etabliert sind. Daraus ergibt sich die Notwendigkeit, in den Anforderungen an einen Generalunternehmer explizit auf nachhaltige Betriebs- und Supportstrukturen sowie auf die Sicherstellung des notwendigen Domänenwissens zu achten und diese als verbindliche Kriterien für die Auswahl zu definieren. Nur so kann gewährleistet werden, dass die angestrebte Modernisierung der Justizinformatik nicht nur technologisch, sondern auch organisatorisch und fachlich nachhaltig umgesetzt wird.

## **8   Fazit**

Das Kapitel Fazit bündelt die wesentlichen Erkenntnisse und Schlussfolgerungen der Konzeptstudie. Aufbauend auf der Analyse der Ausgangslage, der Zielsetzung, den identifizierten Herausforderungen und den erarbeiteten Lösungsansätzen bewertet das Fazit die Tragfähigkeit und Umsetzbarkeit der vorgeschlagenen Architektur- und Betriebsmodelle. Es ordnet die Ergebnisse in den Gesamtkontext der digitalen Transformation der Schweizer Justiz ein, reflektiert die Chancen und Risiken und gibt eine Einschätzung zur Machbarkeit. Abschliessend werden konkrete Empfehlungen für die nächsten Schritte und notwendige Entscheidungen formuliert, um einen realistischen und nachhaltigen Weg für die zukünftige Ausgestaltung der Fachapplikationen aufzuzeigen.

### **8.1   Zusammenfassung**

Die Konzeptstudie zeigt, dass die Schweizer Justiz vor einem grundlegenden Wandel steht. Der Übergang von papiergestützten Verfahren zur vollständig elektronischen Aktenführung betrifft alle Akteure entlang der Justizkette. Die Einführung der eAkte und des elektronischen Rechtsverkehrs (ERV) ist nicht nur eine technologische, sondern auch eine organisatorische und kulturelle Herausforderung. Die bestehenden IT-Systeme sind überwiegend monolithisch aufgebaut, technologisch veraltet und wenig flexibel. Sie erschweren die Integration neuer Funktionen, verursachen hohe Wartungsaufwände und führen zu einer starken Abhängigkeit von wenigen Anbietern.



Die Analyse der Nutzerbedürfnisse und organisatorischen Anforderungen zeigt, dass moderne Fachapplikationen benutzerfreundlich, rechtssicher, effizient und zukunftsfähig sein müssen. Anwender erwarten stabile, intuitive Systeme, die ihre Arbeitsprozesse effektiv unterstützen. Organisationen fordern Standardisierung, Interoperabilität, Skalierbarkeit und Nachhaltigkeit. Die Einbindung professioneller Dritter wie Anwälten und Gutachtern ist entscheidend für Akzeptanz und Effizienz der digitalen Prozesse.

Die Studie empfiehlt eine modulare, ereignisorientierte Architektur mit klar definierten Schnittstellen und Event Sourcing für Revisionssicherheit. Modularisierung und DDD ermöglichen lose gekoppelte Komponenten und flexible Erweiterbarkeit. Ein zentraler Eventhub mit Eventstore stellt die systemweite Kommunikation und Nachvollziehbarkeit sicher. Über Adapter werden spezialisierte Komponenten integriert, was Flexibilität und Investitionsschutz gewährleistet.

Die Marktsituation ist derzeit von einer starken Konzentration auf wenige Anbieter geprägt, die den Justizbereich mit ihren Lösungen dominieren. Diese Konzentration führt zu Abhängigkeiten, erschwert Innovationen und kann die Anpassungsfähigkeit bei zukünftigen Entwicklungen einschränken. Gleichzeitig drängen neue Anbieter mit modernen Technologien auf den Markt, verfügen jedoch häufig nicht über das erforderliche justizspezifische Fachwissen. Die Behörden stehen daher vor der Herausforderung, eigenes Know-how aufzubauen und strategisch zu steuern, um Lock-in-Effekte zu vermeiden und die digitale Souveränität zu stärken. Vor diesem Hintergrund ist auch die Betriebs- und Organisationsform der zukünftigen Systeme sorgfältig zu betrachten.

Für den Betrieb werden drei Modelle beschrieben: Inhouse-Betrieb, Kooperationsmodell und Betrieb durch externe Dienstleister. Als technische Grundlage werden containerbasierte Plattformen empfohlen, da sie Vorteile hinsichtlich Skalierbarkeit, Flexibilität und Wartbarkeit bieten. Diese Empfehlung ist als strategische Orientierung zu verstehen und bedeutet nicht, dass solche Plattformen bereits flächendeckend umgesetzt werden oder zwingend zum Einsatz kommen müssen. Die konkrete Einführung hängt von den individuellen Rahmenbedingungen, dem technischen Reifegrad und den verfügbaren Ressourcen der jeweiligen Organisation ab. In der Praxis nutzen viele Behörden weiterhin klassische IT-Infrastrukturen. Die Transformation hin zu containerbasierten Plattformen erfolgt, wenn überhaupt, schrittweise und bedarfsgerecht.

Da die Entwicklung und Integration moderner Fachapplikationen im Justizbereich eine hohe technische, organisatorische und sicherheitsrelevante Komplexität aufweist, stellt die Beauftragung eines Generalunternehmers einen geeigneten Ansatz dar. Ein erfahrener GU kann die Gesamtverantwortung für Architektur, Integration und Qualitätssicherung übernehmen und die Umsetzung übergreifender Standards sicherstellen. Voraussetzung sind nachgewiesene Erfahrung mit Best-of-Breed-Integrationen, EDA, Sicherheitsstandards, Qualitätssicherung, Know-how-Transfer sowie fundierte Kenntnisse der Schweizer Justiz.

## 8.2 Machbarkeit der vorgeschlagenen Lösung

Die Chancen einer modernen Architektur liegen in der Flexibilität, dem Innovationspotenzial und der Möglichkeit zur schrittweisen Modernisierung. Risiken bestehen vor allem im Integrationsaufwand, in der technischen Komplexität und in der nachhaltigen Pflege von Open-Source-Komponenten. Klare Schnittstellen, verbindliche Standards, gezielter Know-how-Aufbau und funktionierende Governance-Strukturen sind entscheidend, um diese Risiken zu adressieren und die Zukunftsfähigkeit der Lösung zu sichern.

Insgesamt bietet die Studie eine fundierte Entscheidungsgrundlage für die zukünftige Ausgestaltung von Fachapplikationen im Schweizer Justizbereich und skizziert einen umsetzbaren und realistischen Pfad für die digitale Transformation der Justiz.

Gleichzeitig bestehen auch Risiken, die es zu beachten gilt. Die Integration heterogener Systeme und die Umstellung auf eine ereignisorientierte Architektur erfordern erheblichen initialen Aufwand und spezialisiertes Know-how. Nicht alle Organisationen verfügen über die notwendigen personellen und technischen Ressourcen, um die Transformation eigenständig zu bewältigen. Die Vergabe an einen Generalunternehmer kann zu neuen Abhängigkeiten führen, wenn Vertragsgestaltung, Know-how-Transfer und Governance nicht konsequent geregelt werden. Die Umstellung ist nicht nur technisch, sondern auch organisatorisch und kulturell herausfordernd. Veränderungsmanagement und Akzeptanz bei den Anwendern sind erfolgskritisch.

Die Zielsetzung der Studie, eine tragfähige, langfristige und zukunftssichere Lösung für Fachapplikationen im Schweizer Justizbereich zu schaffen, ist grundsätzlich erreichbar. Die vorgeschlagene Architektur und das Betriebsmodell bieten eine solide Basis für eine nachhaltige digitale Transformation. Die Machbarkeit hängt jedoch massgeblich davon ab, wie konsequent Know-how aufgebaut, Governance-Strukturen etabliert und die Umsetzung in realistischen Etappen geplant wird. Die Chancen überwiegen, sofern die identifizierten Risiken aktiv gesteuert werden.

Insgesamt bietet die Studie damit eine fundierte Entscheidungsgrundlage und einen realistischen Pfad für die digitale Transformation der Justiz. Die Umsetzung ist machbar, erfordert jedoch strategische Planung, klare Verantwortlichkeiten und die Bereitschaft, in Know-how und Governance zu investieren.

### 8.3 Nächste Schritte

Ein möglicher Pfad zur weiteren Konkretisierung könnte darin bestehen, auf Basis der vorliegenden Architektur- und Betriebsmodelle konkrete Umsetzungsprojekte zu definieren, die in klar abgegrenzten Etappen erfolgen. Zunächst sollten Pilotierungen in ausgewählten Organisationseinheiten durchgeführt werden, um Erfahrungen mit der modularen Architektur. Parallel dazu ist es notwendig, verbindliche Standards für Schnittstellen, Datenmodelle und Governance-Strukturen zu entwickeln und abzustimmen. Die Auswahl und Qualifizierung von Generalunternehmern oder Technologiepartnern sollte transparent und anhand der in der Studie definierten Anforderungen starten. Ebenso wichtig ist es, frühzeitig gezielte Schulungs- und Weiterbildungsmaßnahmen für die beteiligten Behörden und deren Mitarbeitende zu planen und umzusetzen, damit das notwendige technische und organisatorische Know-how aufgebaut und nachhaltig verankert werden kann. Im weiteren Verlauf müssen die gewonnenen Erkenntnisse aus den Pilotierungen systematisch ausgewertet und in die flächendeckende Umsetzung überführt werden. Die notwendigen Entscheidungen betreffen insbesondere die Priorisierung der Umsetzungsschritte, die Ressourcenplanung, die Ausgestaltung der Governance sowie die Festlegung von Qualitäts- und Sicherheitsstandards. Nur durch eine schrittweise, strukturierte und partizipative Vorgehensweise kann die digitale Transformation der Justiz nachhaltig und erfolgreich gestaltet werden.