IT-PLANUNGSRAT

Frankfurt am Main
29.05.2026

Zielarchitektur föderale API-Autorisierungsinfrastruktur

- Langfassung -

Konzeption einer Basisinfrastruktur zur Realisierung eines durchgängigen föderalen Plattform-Ökosystems

Inhaltsverzeichnis

1 Einleitung

Die föderale Verwaltung steht vor der Herausforderung, eine wachsende Zahl digitaler Basisdienste sicher, einheitlich und wirtschaftlich miteinander zu verknüpfen. Dieses Kapitel erläutert, warum ein gemeinsames Fundament für die API-Autorisierung notwendig ist und was die Zielarchitektur zur föderalen API-Autorisierung dazu beiträgt. Kapitel 1.1 beschreibt die Ausgangslage, die dem Vorhaben zugrunde liegt, und ordnet es in den Kontext des IT-Planungsrats ein. Kapitel 1.2 legt den fachlichen Umfang der Zielarchitektur anhand der relevanten Wertströme dar. Kapitel 1.3 erläutert das Ziel des Dokuments und seinen Verwendungszweck.

1.1 Projektkontext und Ausgangslage

Die föderale IT der öffentlichen Verwaltung wächst kontinuierlich: Basisdienste (z. B. Postfächer, FIT-Connect, Bezahldienste, NOOTS-Komponenten) bilden zunehmend das technische Rückgrat digitaler Verwaltungsleistungen. Onlinedienste, Fachverfahren und Unternehmensanwendungen müssen sich gegenüber diesen Basisdiensten autorisieren – und das nach unterschiedlichen Vorgaben, mit unterschiedlichen Registrierungsprozessen und ohne gemeinsame Standards für API-Absicherung und Berechtigungsmanagement. Daraus entstehen konkrete Probleme:

- **Mehrkosten und Komplexität in der Implementierung:** Jeder Basisdienst erfordert eine eigene Integrationslogik, da keine einheitlichen technischen Muster existieren. Betreiber von Onlinediensten und Fachverfahren müssen für jeden Basisdienst gesonderte Autorisierungsmechanismen entwickeln und warten.
- **Mehrkosten bei der Registrierung:** Das Fehlen automatisierter, standardisierter Registrierungsprozesse zwingt Betreiber zu manuellen, basisdienst-spezifischen Onboarding-Verfahren, was den Integrationsaufwand erheblich erhöht.
- **Sicherheitsrisiken:** Ohne gemeinsame Sicherheitsvorgaben entstehen heterogene, teils unzureichend gesicherte Autorisierungslösungen, die die Gesamtsicherheit der föderalen IT-Landschaft gefährden.
- **Behinderung eines API-First-Ökosystems:** Die fehlende Standardisierung erschwert die skalierbare Nachnutzung von Basisdiensten und bremst die Entwicklung eines durchgängig interoperablen föderalen API-Ökosystems.

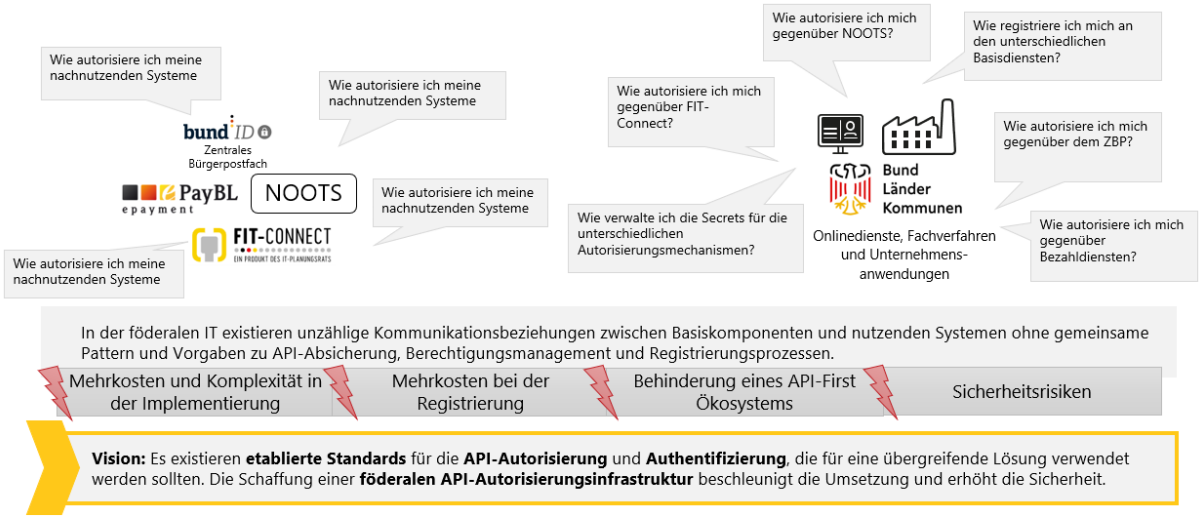


Abbildung 1: Ausgangslage

Das Projekt adressiert diese Ausgangslage und schafft das notwendige gemeinsame Fundament. Eine erste Pilotierung standardisierter API-Autorisierung im Kontext der Bezahldienste lieferte dabei praktische Erkenntnisse, die in das Vorhaben eingeflossen sind.

Das Projekt „Föderale API-Autorisierungsinfrastruktur" (Projekt-ID: itPLR-25-004) wurde durch Beschluss 2025/22 der 47. Sitzung des IT-Planungsrats vom 26. Juni 2025 initialisiert. Es ist dem Schwerpunktthema „Digitale Transformation" zugeordnet und adressiert dort das Zielbild der Standardisierung. Antragstellendes Bundesland war Sachsen-Anhalt; die Projektergebnisse wurden zur Sommersitzung 2026 des IT-Planungsrats eingereicht. Die Laufzeit betrug 11 Monate von Juli 2025 bis Juni 2026.

Das Vorhaben umfasst zwei zentrale Liefergegenstände, die – wie die nachfolgende Abbildung zeigt – in ihrem Geltungsbereich bewusst unterschieden werden.

Die **Sicherheitsvorgaben** definieren auf Basis des FAPI 2.0-Standards ein einheitliches Sicherheitsniveau für die API-Absicherung mit OAuth und OpenID Connect. Ihr Geltungsbereich umfasst alle APIs der öffentlichen Verwaltung.

Die **Zielarchitektur** – das vorliegende Dokument – beschränkt sich auf die APIs föderaler Basisdienste und legt dar, wie Betreiber, Integratoren und Nutzer in einer föderalen Autorisierungsinfrastruktur skalierbar, sicher und wirtschaftlich zusammenwirken können. Die Zielarchitektur unterliegt dabei selbst den föderalen Sicherheitsvorgaben. Beide Liefergegenstände werden durch unterstützende Artefakte flankiert: Anforderungslisten, Architekturziele, Architekturprinzipien und Architekturentscheidungen (ADRs). Als optionale künftige Ergebnisse sind

eine technische Referenzarchitektur sowie eine darauf aufbauende Referenzimplementierung vorgesehen.

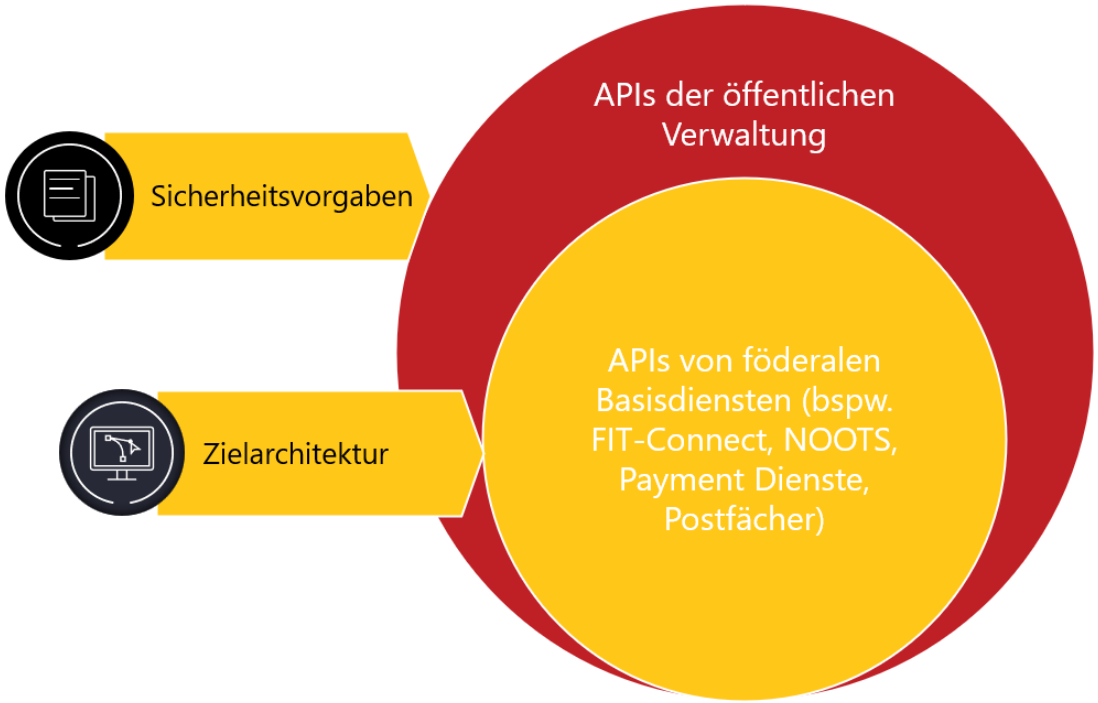


Abbildung 2: Fokus der Projektliefergegenstände

1.2 Umfang und Fokus der Zielarchitektur

Der Umfang der Zielarchitektur wird über eine Betrachtung der relevanten Wertströme definiert. Ein Wertstrom beschreibt dabei die Abfolge von Aktivitäten, die notwendig sind, um einem bestimmten Stakeholder einen konkreten Nutzen zu liefern. Die nachfolgende Abbildung zeigt die drei identifizierten Wertströme im Überblick; eine detaillierte Beschreibung folgt in Kapitel 3.1.

Aus der in Kapitel 1.1 beschriebenen Ausgangslage ergeben sich drei komplementäre Wertströme, die das Vorhaben adressiert:

APIs von Basisdiensten bereitstellen: Betriebsorganisationen von Basisdiensten können ihre APIs auf standardisierte Weise im föderalen Kontext verfügbar machen – inklusive der Beschreibung von Berechtigungsmodellen und der Veröffentlichung von Kataloginformationen. Das Ergebnis: APIs des Basisdienstes sind föderal nutzbar.

APIs von Basisdiensten in fachliche Anwendungen integrieren: Betriebsverantwortliche Stellen und Software-Lieferanten können die benötigten APIs von Basisdiensten über einheitliche Prozesse beantragen, ihre Software registrieren und API-Clients automatisiert an allen relevanten Basisdiensten anmelden. Das Ergebnis: Alle relevanten APIs wurden in die eigene Lösung integriert.

Angebote von Basisdiensten nutzen: Basisdienstnutzer können zur Laufzeit sicher und nachvollziehbar auf Basisdienste zugreifen – unabhängig davon, welcher Basis-dienst genutzt wird. Das Ergebnis: Basisdienste können genutzt werden und die Nutzung bleibt transparent.

Die drei Wertströme sind komplementär zueinander: Erst ihr abgestimmtes Zusammenspiel schafft ein durchgängig anschlussfähiges föderales API-Ökosystem.

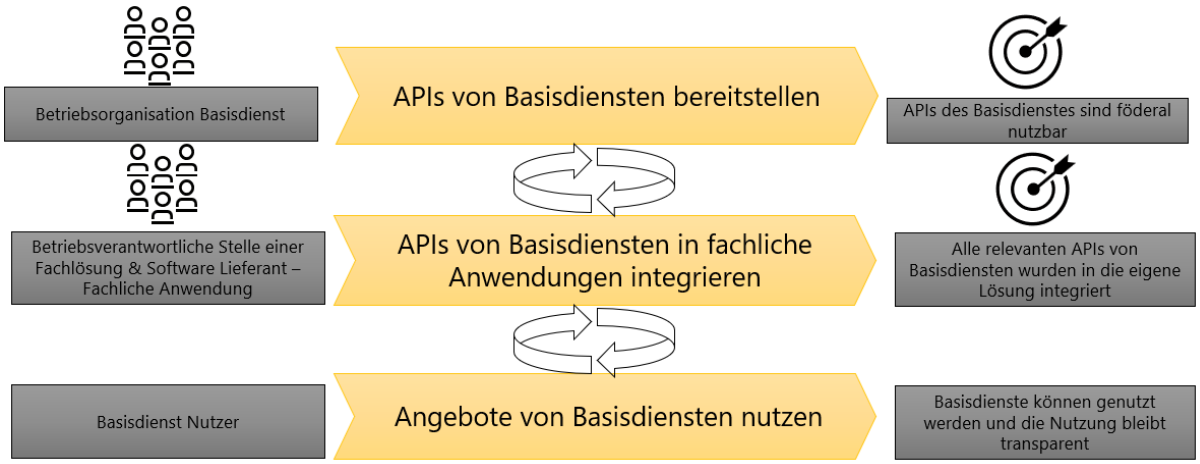


Abbildung 3: Umfang und Fokus der Zielarchitektur

1.3 Ziel des Dokuments

Dieses Dokument dient dem IT-Planungsrat als Entscheidungsgrundlage für den Aufbau einer föderalen API-Autorisierungsinfrastruktur. Es legt dar, wie eine solche Infrastruktur konzeptionell aussehen kann, welche organisatorischen und technischen Voraussetzungen für ihre Umsetzung erforderlich sind, welche Migrationsaufgaben auf die betroffenen Basisdienste und Betreiberorganisationen zukommen und welche Fragen im Rahmen dieses Konzepts noch offengeblieben sind und einer weiteren Klärung bedürfen.

Auf Basis dieses Dokuments soll der IT-Planungsrat ein Umsetzungsprojekt beschließen können, das die identifizierten Implementierungs- und Migrationsaufgaben durchführt und die föderale API-Autorisierungsinfrastruktur in den produktiven Betrieb überführt.

2 Strategische Ausrichtung

Die strategische Ausrichtung beschreibt den übergeordneten Rahmen, der alle nachfolgenden Architektur- und Designentscheidungen der föderalen API-Autorisierungsinfrastruktur leitet. Sie beantwortet die Frage, wohin das Vorhaben führen soll, welche Ziele dabei im Vordergrund stehen und nach welchen Prinzipien gehandelt wird.

Kapitel 2.1 formuliert die Architekturvision als prägnantes Zielbild, das den angestrebten Zielzustand aus Sicht der relevanten Stakeholder beschreibt. Kapitel 2.2 konkretisiert diese Vision in Form projektspezifischer Architekturziele, die den Nutzen der Infrastruktur für Betreiber, Integratoren und die föderale IT-Landschaft insgesamt operationalisieren. Kapitel 2.3 legt die Architekturprinzipien fest, die als verbindliche Leitlinien bei der Umsetzung der Zielarchitektur anzuwenden sind.

2.1 Architekturvision

Folgendes Vision Statement richtet als Nordstern die Konzeption der Zielarchitektur aus:

> *„Föderale Basisdienste sind nach einheitlichen Sicherheitsstandards absicherbar und wirtschaftlich integrierbar – unabhängig davon, wer sie betreibt oder nutzt. Eine gemeinsame Autorisierungsinfrastruktur schafft das technische Fundament für ein interoperables föderales Plattform-Ökosystem, in dem Mehrfachaufwände entfallen und Berechtigungen nachvollziehbar geregelt sind. Sie steht öffentlichen IT-Dienstleistern und privaten Unternehmen gleichermaßen offen – und wird so zur Grundlage eines lebendigen GovTech-Ökosystems, das Innovationen in der Verwaltungsdigitalisierung erst möglich macht."*

2.2 Projektspezifische Architekturziele

Diese Vision Statement ist auch die Basis für die projektspezifischen Architekturziele und Architekturprinzipien, die strategische Ausrichtung des Projekts weitere operationalisieren. Eine Übersicht der acht Ziele findet sich in folgender Abbildung.

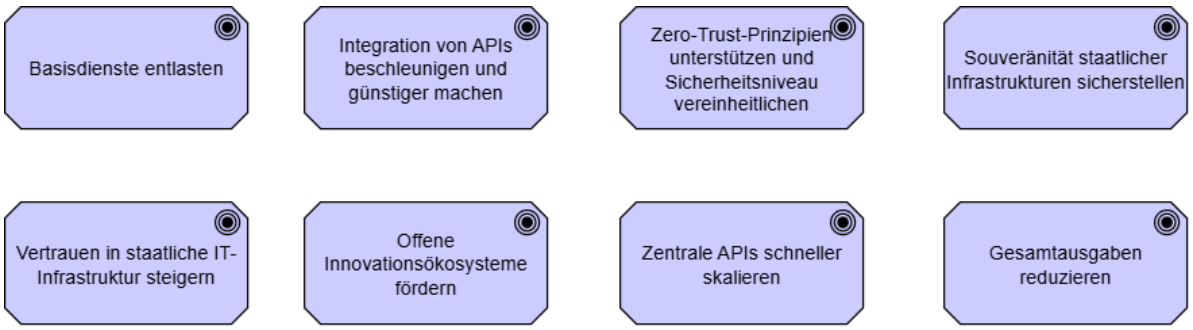


Abbildung 4: Projektspezifische Architekturziele

- **Basisdienste entlasten:** Die föderale API-Autorisierungsinfrastruktur übernimmt technische Querschnittsfunktionen, die Basisdienst-Betreiber heute selbst implementieren müssen. Dazu gehören insbesondere Registrierungsprozesse, Token-Management und die Durchsetzung von Berechtigungsregeln. Durch diese Bündelung können sich Basisdienst-Teams auf ihre fachlichen Kernaufgaben konzentrieren. Die Infrastruktur schafft so eine nachhaltige Entlastung, ohne die Eigenverantwortung der Basisdienste für ihre fachliche Berechtigungslogik einzuschränken.
- **Integration von APIs beschleunigen und günstiger machen:** Standardisierte Registrierungsprozesse und automatisierte Anbindungsverfahren reduzieren den Zeit- und Kostenaufwand für die Integration föderaler Basisdienste erheblich. Statt für jeden Basisdienst eigene Integrationslogik zu entwickeln, nutzen alle Anwendungen dieselben Muster und Schnittstellen. Dies senkt die Einstiegshürde für neue Integratoren und ermöglicht eine breitere Nachnutzung von Basisdiensten. Auch kleine Entwicklungsorganisationen und Startups können so effizient integrieren.
- **Zero-Trust-Prinzipien unterstützen und Sicherheitsniveau vereinheitlichen:** Die Infrastruktur setzt Zero-Trust konsequent um – kein System, kein Netzwerkbereich und kein Akteur gilt ohne explizite Verifikation als vertrauenswürdig. Jede Anfrage wird unabhängig von Netzwerkherkunft oder institutioneller Zugehörigkeit authentifiziert und autorisiert. Einheitliche Sicherheitsvorgaben auf Basis von FAPI 2.0 und OAuth 2.0 ersetzen heterogene, teils unzureichend gesicherte Einzellösungen. Das Gesamtsicherheitsniveau der föderalen IT-Landschaft wird damit systematisch und nachhaltig angehoben.
- **Souveränität staatlicher Infrastrukturen sicherstellen:** Die föderale Autorisierungsinfrastruktur basiert auf offenen Industriestandards und Open-Source-Komponenten, um Abhängigkeiten von einzelnen Herstellern zu vermeiden. Dezentrale Betriebsmodelle stellen sicher, dass keine einzelne Stelle eine unkontrollierte Kontrolle über kritische

Sicherheitsfunktionen erlangen kann. Die Nutzung etablierter Standards ermöglicht es, Komponenten bei Bedarf auszutauschen oder weiterzuentwickeln. Staatliche IT-Souveränität wird so durch Architektur gesichert, nicht durch Vertrauen in einzelne Akteure.

- **Vertrauen in staatliche IT-Infrastruktur steigern:** Transparente, nachvollziehbare und revisionssichere Protokollierung aller Berechtigungsentscheidungen schafft Vertrauen bei Bürgerinnen, Bürgern und Unternehmen. Die Infrastruktur macht sicherheitsrelevante Ereignisse für autorisierte Prüfinstanzen und Auditoren einsehbar. Berechtigungen basieren auf klar definierten Regeln und Prozessen – nicht auf implizitem institutionellem Vertrauen. Diese Transparenz stärkt das gesellschaftliche Vertrauen in eine sichere und rechtskonforme Verwaltungs-IT.
- **Offene Innovationsökosysteme fördern:** Durch standardisierten, diskriminierungsfreien Zugang zur Infrastruktur können öffentliche IT-Dienstleister und private Unternehmen gleichermaßen föderale Basisdienste in ihre Lösungen integrieren. Startups und etablierte Unternehmen finden dieselben technischen Voraussetzungen vor und können auf einem Level Playing Field innovieren. Die Infrastruktur wird zur offenen Plattform für ein lebendiges GovTech-Ökosystem. Neue Lösungen für die Verwaltungsdigitalisierung entstehen schneller, weil das technische Fundament nicht mehr selbst aufgebaut werden muss.
- **Zentrale APIs schneller skalieren:** Neue Basisdienste können durch die standardisierte Infrastruktur deutlich schneller in das föderale Ökosystem eingebunden und skaliert werden. Einmalig definierte Sicherheitsvorgaben, Registrierungsprozesse und Berechtigungslogiken gelten für alle Basisdienste ohne Mehraufwand. Die Nachnutzung bestehender Infrastrukturkomponenten reduziert die Time-to-Market für neue API-Angebote erheblich. Das föderale Plattform-Ökosystem wächst damit in einem Tempo, das ohne gemeinsame Infrastruktur nicht erreichbar wäre.
- **Gesamtausgaben reduzieren:** Die Bündelung von Querschnittsfunktionen in einer gemeinsamen Infrastruktur vermeidet teure Mehrfachentwicklungen bei Bund, Ländern und Basisdienst-Betreibern. Standardisierte Komponenten und Prozesse senken den Betriebsaufwand durch Skaleneffekte und spezialisiertes Know-how. Automatisierte Anbindungsprozesse reduzieren manuelle Prüf- und Onboarding-Aufwände auf beiden Seiten. Langfristig sinken die Gesamtbetriebskosten der föderalen IT-Landschaft durch eine konsolidierte und wirtschaftlich betriebene Autorisierungsinfrastruktur.

2.3 Projektspezifische Architekturprinzipien

Die projektspezifischen Architekturprinzipien bilden den verbindlichen Handlungsrahmen für alle Architektur- und Designentscheidungen im Rahmen der föderalen API-Autorisierungsinfrastruktur. Sie konkretisieren das Vision Statement und die Architekturziele auf der Ebene von Gestaltungsregeln, die bei der Auswahl von Technologien, der Konzeption von Prozessen und der Bewertung von Lösungsalternativen anzuwenden sind. Die nachfolgende Tabelle listet alle geltenden Architekturprinzipien mit ihrer ID, ihrer Bezeichnung und einer kompakten Erklärung auf. Die vollständigen Beschreibungen einschließlich Begründungen, Abhängigkeiten und Auswirkungen sind im Projektrepository auf OpenCode verfügbar:

https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/tree/main/Architekturprinzipien

Tabelle 1: Projektspezifische Architekturprinzipien

ID	Name	Erklärung
API-Infra-P-001	Offene Industriestandards und Best Practices nutzen	Die föderale Infrastruktur nutzt für Funktionen, Datenmodelle, Schnittstellen und Architekturmuster im Bereich der API-Absicherung, wo möglich, offene Industriestandards und Best-Practices, die sich fest etabliert haben. Auch neu aufkommende Entwicklungen werden genutzt, sofern sie sich in ein bestehendes Ökosystem aus Industriestandards und Best Practices einfügen lassen und eine Etablierung innerhalb dieses Ökosystems denkbar ist.
API-Infra-P-002	Individualentwicklungen und verwaltungsspezifische Lösungen vermeiden	Existierende Standardlösungen, die für eine Vielzahl von Nutzern über mehrere Branchen und Einsatzbereiche hinweg entwickelt und unterstützt werden, sollen grundsätzlich Vorrang gegenüber Individualentwicklungen haben. Verwaltungsspezifische Lösungen, die nur für den öffentlichen Sektor entwickelt werden, sollen ebenfalls vermieden werden, sofern sie verwaltungsunspezifische Einsatzbereiche und Anforderungen, wie beispielsweise Identitätsmanagement, Kryptografie oder Token-Management, adressieren. Individualentwicklungen oder verwaltungsspezifische Lösungen sollen nur dann genutzt werden, wenn damit nicht vermeidbare, hochspezifische Anforderungen adressiert werden, die für die föderale API-Autorisierungsinfrastruktur zu erfüllen sind, für die aber kein Bedarf in anderen Branchen und Einsatzbereichen zu erwarten ist.
API-Infra-P-003	Etablierte Verwaltungsinfrastruktur als zulässige Standardlösung berücksichtigen	Etablierte Komponenten der digitalen Verwaltungsinfrastruktur, die breit im öffentlichen Sektor genutzt werden, einer klaren Governance unterliegen und auf offenen Industriestandards basieren, sind als zulässige Standardlösungen für die föderale API-Autorisierungsinfrastruktur zu berücksichtigen. Sie sind nicht als zu vermeidende verwaltungsspezifische Lösungen im Sinne von Prinzip 002 einzuordnen. Die Nutzung solcher Verwaltungsinfrastruktur ist zulässig, sofern sie interoperabel ausgestaltet ist, sich in bestehende Standardökosysteme einfügt und keine dauerhafte Bindung an proprietäre oder fachverfahrensspezifische Insellösungen entsteht.
API-Infra-P-004	Verwaltungsspezifische Anforderungen in	Bestehen fachliche oder sicherheitstechnische Anforderungen der föderalen API-Autorisierungsinfrastruktur, die durch keinen bestehenden offenen Standard und kein geeignetes Produkt abgedeckt werden und

ID	Name	Erklärung
	Standardisierungsprozesse einbringen	über eine einzelne Implementierung hinaus Relevanz besitzen, sollen diese aktiv in geeignete Standardisierungsprozesse eingebracht werden. Eigenentwicklungen zur Abdeckung solcher Anforderungen sind nur als zeitlich befristete Übergangslösungen zulässig, sofern ein klarer Standardisierungsweg identifiziert ist und aktiv verfolgt wird.
API-Infra-P-005	„Never trust, always verify" bei allen Zugriffen	Wenn ein System auf ein anderes System zugreift, muss das antwortende System das anfragende System authentifizieren und dessen Autorisierungen sowie entscheidungsrelevante Rahmenbedingungen bzw. Informationen verifizieren. Ein Beispiel für einen solchen Zugriff ist ein API-Client, der auf die API eines Basisdienstes zugreift, oder der Basisdienst, der auf den Authorization Server zugreift. Für besonders kritische Informationen exponierter Systeme sollten stets ergänzende Informationsquellen zu Veränderungstransaktionen vorhanden sein. Diese ermöglichen es zumindest den nachnutzenden Systemen oder Risikobewertungssystemen, besondere Risikosituationen zu erkennen, die durch ungewöhnliche Änderungen an Datenbeständen verursacht wurden, und weitere Validierungs- und Schutzmaßnahmen einzuleiten.
API-Infra-P-006	Minimale Zugriffsberechtigungen und dynamische Zugriffsüberprüfungen	Alle Systeme im Kontext der angedachten föderalen Infrastruktur sollen nur die Berechtigungen und Autorisierungen erhalten, die sie zwingend für ihre Aufgabe benötigen („Least-Privilege-Prinzip"). Der tatsächliche Umfang der Zugriffsberechtigung auf Basis dieser Minimalberechtigungen soll im Sinne des Policy-based Access-Prinzips davon abhängen, wie vertrauenswürdig das System angesichts der Rahmenbedingungen des Zugriffs und der dafür definierten Zugriffsregelungen ist. Zu diesen Rahmenbedingungen zählen beispielsweise die Uhrzeit des Zugriffs und die Anzahl der Zugriffe über einen bestimmten Zeitraum hinweg in der gesamten Infrastruktur.
API-Infra-P-007	Nachvollziehbarkeit und Auditierbarkeit aller relevanten Systemaktivitäten	Die föderale Infrastruktur soll so konzipiert werden, dass alle relevanten Ereignisse und Rahmenbedingungen für Abrufe protokolliert werden können. Zudem sollen die Protokollierungen so umgesetzt werden, dass Manipulationen entgegengewirkt wird, indem beispielsweise mehrere Datenpunkte für ein jeweiliges Ereignis oder einen Prozess vorgesehen sind, die redundant gespeichert werden. Im Sinne dieses Prinzips sollen Informationen auch über die Grenzen einzelner Teilnehmer hinweg geteilt werden, um eine vollständige Transparenz über alle relevanten Bereiche zu erreichen und somit die Sicherheit von API-Zugriffen zu gewährleisten. Informationen, deren öffentliches Bekanntwerden nicht der Sicherheit des Gesamtsystems oder den Datenschutzrechten entgegensteht, sollen allen Teilnehmern der Infrastruktur und auch

ID	Name	Erklärung
		Dritten bereitgestellt werden. Bei besonders kritischen Ereignissen, die einzelne Teilnehmer betreffen, sind diese Teilnehmer über diese Ereignisse und die dazugehörigen Rahmenbedingungen zu informieren, damit sie im Falle eines Angriffs aktiv reagieren können.
API-Infra-P-008	Dezentralität und Redundanz von Sicherheitsmechanismen	Die übergreifende Architektur soll so konzipiert sein, dass möglichst viele essenzielle Absicherungsfunktionen (wie z. B. Token-Management), die für die Funktionsfähigkeit von APIs und die Sicherheit der Gesamtinfrastruktur wichtig sind, als dezentrale Komponenten betrieben werden können. Zudem sollten für den Schutz eines spezifischen Angriffsziels unterschiedliche Sicherheitsmechanismen auf mehrere Komponenten verteilt werden, damit die Kompromittierung einer einzelnen Komponente keinen erfolgreichen Angriff ermöglicht.
API-Infra-P-009	Open-Source-Priorisierung für kritische Komponenten	Wenn verfügbare Open-Source-Lösungen für kritische Komponenten der föderalen Infrastruktur eine ähnliche Eignung aufweisen wie proprietäre Lösungen, sind Open-Source-Lösungen vorzuziehen. Dabei ist darauf zu achten, dass Open-Source-Lösungen von einem breiteren Ökosystem getragen werden und nicht lediglich verwaltungseigene Lösungen oder Lösungen einzelner Unternehmen mit einer Open-Source-Lizenz versehen sind. Zudem sollte darauf geachtet werden, dass ein Ökosystem bereitsteht, das den notwendigen Support für den Einsatz der Lösung gewährleisten kann.
API-Infra-P-010	Bündelung technischer Querschnittsfunktionen	Technische Querschnittsfunktionen, die von vielen Fachlösungen oder Basisdiensten in gleicher Weise benötigt werden (beispielsweise im Bereich der API-Sicherung), sollen nach Möglichkeit in spezialisierten Komponenten gebündelt und über lose gekoppelte Schnittstellen an die Fachlösungen bereitgestellt werden. Die Betreuung dieser Komponenten soll durch spezialisierte Teams übernommen werden. Dieses Prinzip impliziert nicht zwingend eine zentral betriebene Komponente, sondern kann auch eine zentral entwickelte/eingekaufte Komponente mit zentralem Support sein. Diese läuft dann beispielsweise als Sidecar in der lokalen Betriebsumgebung der Fachlösung oder des Basisdienstes.
API-Infra-P-011	Flexibilität und Anpassbarkeit API-spezifischer Berechtigungsmodelle	Die föderale API-Autorisierungsinfrastruktur soll gewährleisten, dass Basisdienste die berechtigungsrelevanten Informationen ihrer APIs weiterhin flexibel an ihre fachlichen Anforderungen anpassen können.

ID	Name	Erklärung
API-Infra-P-012	Automatisierung von Anbindungsprozessen	Alle Prozessschritte im Anbindungsprozess – von der initialen Registrierung eines Softwareentwicklers oder Betreibers bis hin zur Registrierung der konkreten Software an der gewünschten API – sind vollständig zu automatisieren, sofern hierfür sichere und technisch praktikable Lösungen vorliegen. Automatisierung bedeutet in diesem Zusammenhang, dass die anzubindende Lösung bzw. Betriebsumgebung die Prozessschritte zur Anbindung über spezifische APIs und sonstige technische Mechanismen automatisiert umsetzen kann. In der Regel bedeutet dies, dass weiterhin eine Person auf der Seite der anzubindenden Lösung verantwortlich ist, die Prozesse in ihrem System jedoch lediglich auslöst bzw. überwacht.
API-Infra-P-013	Ermöglichung effizienter und unterbrechungsfreier Backoffice-Verwaltungsprozesse	Die föderale API-Autorisierungsinfrastruktur soll so gestaltet werden, dass effiziente und unterbrechungsfreie Backoffice-Verwaltungsprozesse aus Sicht der Verwaltung gewährleistet sind. Es soll vermieden werden, dass die Bearbeitung von Backoffice-Verwaltungsprozessen beispielsweise durch manuelle Freigaben einzelner Nutzender unterbrochen oder verzögert wird.
API-Infra-P-014	Wiederverwendung und Bündelung vor Neuentwicklung	Für alle benötigten Funktionen sollte grundsätzlich geprüft werden, ob bereits geeignete föderale Lösungen bei vorhandenen Basisdiensten vorliegen, die nachgenutzt bzw. für die ermittelten Anforderungen erweitert oder weiterentwickelt werden können. Eine solche Nachnutzung kann von der Nutzung einer zentral betriebenen Serverlösung bis hin zur Nachnutzung der entwickelten oder eingekauften Software reichen. Falls keine geeigneten Lösungen vorliegen oder diese sich nicht weiterentwickeln lassen, sollte bei Neuentwicklungen geprüft werden, ob sich die Bedarfe anderer Basisdienste bündeln lassen, um eine Konsolidierung der IT-Landschaft zu ermöglichen.
API-Infra-P-015	Vertrauen basiert auf Protokollen und Prozessen, nicht auf Institutionen	Grundsätzlich wird davon ausgegangen, dass die Sicherheit und Vertrauenswürdigkeit der föderalen API-Autorisierungsinfrastruktur durch Protokolle und Prozesse gewährleistet wird. Es wird jedoch nicht davon ausgegangen, dass sich die beteiligten Institutionen immer an geltende Vorgaben und Gesetze halten.
API-Infra-P-016	Personenbezogene Daten minimieren	Das Prinzip bedeutet, dass in der Architektur nur die unbedingt erforderlichen personenbezogenen Daten erhoben, verarbeitet und gespeichert werden. Es basiert auf dem Grundsatz der Datenminimierung aus der DSGVO, der besagt: Personenbezogene Daten müssen dem Zweck angemessen und erheblich sowie auf das für die Zwecke der Verarbeitung notwendige Maß beschränkt sein.

ID	Name	Erklärung
API-Infra-P-017	Dynamische Berechtigungssteuerung	Das Prinzip besagt, dass die Architektur so gestaltet sein muss, dass Zugriffsrechte sofort und flexibel geändert oder entzogen werden können – ohne Verzögerung und ohne technische Hürden. Berechtigungen dürfen nicht statisch oder fest verdrahtet sein, sondern müssen über zentrale oder verteilte Mechanismen kontrollierbar bleiben.

3 Geschäftsarchitektur

Die Geschäftsarchitektur liefert einen Überblick über den fachlichen Kontext, der ein gemeinsames Verständnis der organisatorischen Zusammenhänge ermöglicht und dazu dient, strategische Ziele und fachliche Anforderungen aufeinander abzustimmen. Der Abschnitt gliedert sich in folgende Bereiche:

- Wertstrombetrachtung
- Übersicht der Informationskonzepte
- Ableitung der benötigten strategischen Fähigkeiten
- Prozessübersicht

3.1 Wertstrombetrachtung

Ein Wertstrom stellt die Ende-zu-Ende-Abfolge von Aktivitäten dar, die aufgrund eines stakeholdergetriebenen Auslösers (Ein Bedarf eines Stakeholders oder ein wichtiges Ereignis für diesen Stakeholder) einen Wert für diesen Stakeholder generieren. Ein Wertstrom ist konzeptionell entlang eines Geschäftsprozesses modelliert, aber fokussiert sich auf Schritte der Wertgenerierung für den zentralen internen oder externen Stakeholder, während Geschäftsprozesse eher nach operativen Gesichtspunkten (bspw. organisatorische Übergabepunkte oder zeitliche Abläufe) strukturiert sind. Supportprozesse wie Sperrungen, Löschungen oder Störungsbehandlungen gehören nicht zum Wertstrom, da sie keinen direkten Wert für den Stakeholder erzeugen.

Damit soll sichergestellt werden, dass die Geschäftsarchitektur und damit die darauf aufbauende IT-Architektur auf die Wertgenerierung für Stakeholder ausgerichtet wird, anstatt existierende operative Strukturen und Abläufe zu replizieren. Zugleich definieren die betrachteten Wertströme den Umfang bzw. Scope der Architektur, da sich alle weiteren Konzeptionen und Betrachtungen an den Wertströmen ausrichten, weshalb die Wertströme schon in Abschnitt zum Umfang und Fokus der Zielarchitektur beschrieben wurden. (siehe Kapitel 1.2).

Für die Zielarchitektur „Föderale API-Autorisierungsinfrastruktur“ wurden drei zentrale Wertströme identifiziert, die im Folgenden näher beschrieben werden:

- APIs von Basisdiensten integrieren
- APIs von Basisdiensten bereitstellen
- Angebote von Basisdiensten nutzen

Aufgrund der zentralen Bedeutung für den Wertstrom werden die auslösenden Stakeholder (die Stakeholder, für die ein Wert generiert wird) näher beschrieben. Die anderen im Wertstrom beschriebenen Stakeholder werden im Rahmen der Übersicht der Informationskonzepte (siehe Kapitel 3.2) nochmals detaillierter definiert und in Beziehung gebracht.

3.1.1 Wertstrom „APIs von Basisdiensten in fachliche Anwendungen integrieren“

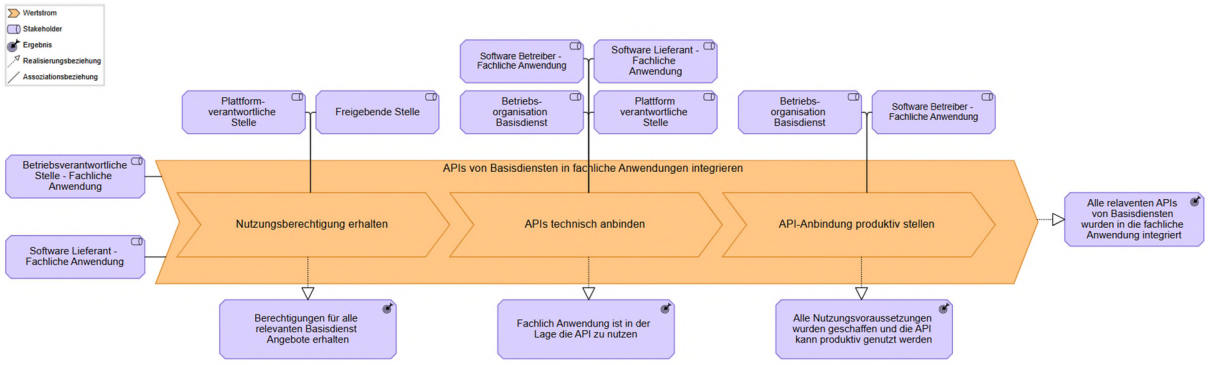


Abbildung 5: Wertstrom APIs von Basisdiensten integrieren

Tabelle 2: Steckbrief Wertstrom "APIs von Basisdiensten integrieren"

Name	APIs von Basisdiensten bereitstellen
Kurzbeschreibung	Dieser Wertstrom beinhaltet alle Aktivitäten zur Bereitstellung von APIs und damit verbundenen Angeboten (Zentrale Frontendanwendungen auf Basis dieser APIs) von Basisdiensten in einem föderalen Plattform-Ökosystem.
Auslösender Stakeholder	• Betriebsorganisation Basisdienst: Ist die für einen Basisdienst verantwortliche Stelle, die alle Entscheidungen in Bezug auf die Entwicklung und den Betrieb eines Basisdienstes trifft. • Fachverbundverantwortliche Stelle: Ist die für einen Fachverbund verantwortliche Stelle, die einen Fachverbund steuert, der wiederum mehrere Basisdienste und Standards für einen Kontext (bspw. NOOTS oder Innenverwaltung) bündelt und koordiniert.

Wertversprechen	Das zentrale Wertversprechen besteht darin, dass APIs von Basisdiensten effektiv allen nachnutzenden Stellen bereitgestellt werden können und die verantwortlichen Stellen beim Bereitstellungsprozess entlastet werden. Durch die übergreifende Koordinierung der Bereitstellung wird sichergestellt, dass alle Angebote fachlich aufeinander abgestimmt sind und mit konsistenten Prozessen und Berechtigungslogiken bereitgestellt werden.

Tabelle 3: Taskbeschreibung "APIs von Basisdiensten integrieren"

Wertstromschritt	Kurzbeschreibung	Ergebnis	Beteiligte Stakeholder
Bereitstellungsberechtigung für föderale Plattformangebote erhalten	Die Aktivitäten beinhalten die Beantragung, Prüfung und Gewährung von Bereitstellungsberechtigungen von föderalen Plattformangeboten.	Die Betriebsorganisation Basisdienst oder die Fachverbundverantwortliche Stelle darf ihre Angebote föderal bereitstellen.	Fachverbundverantwortliche Stelle, Plattformverantwortliche Stelle
Basisdienstangebot und Berechtigungsmodell festlegen	Die Aktivitäten beinhalten die Abstimmung des Basisdienstangebots mit bestehenden Angeboten anderer Basisdienste, die Festlegung des Berechtigungsmodell auf grobgranularer Zugriffsebene und der Integration des Berechtigungsmodells in die übergreifende Berechtigungsstruktur.	Alle Informationen zum Basisdienstangebot und dem Berechtigungsmodell sind spezifiziert und alle Grundlagen für die Berechtigungsprüfung auf grobgranularer Zugriffsebene sind implementiert und getestet.	Fachverbundverantwortliche Stelle, Plattformverantwortliche Stelle, Freigebende Stelle
Basisdienstangebot veröffentlichen	Die Aktivitäten beinhalten die Veröffentlichung von allen Informationen, die notwendig sind um die Angebote eines Basisdienstes zu nutzen (bspw. Technische Adressen von Betriebsumgebungen, Zertifikate, Voraussetzung für die Nutzung) sowie die Produktivsetzung von Integrationen des Basisdienstes in übergreifende Strukturen (bspw. Zur Registrierung von API-	Alle Informationen zur Nutzung des Basisdienstangebots sind bereitgestellt und eine produktive Nutzung des Angebots ist möglich.	Plattformverantwortliche Stelle

	Consumern und dem allgemeinen Berechtigungsmanagement)		

3.1.2 Wertstrom „APIs von Basisdiensten bereitstellen“

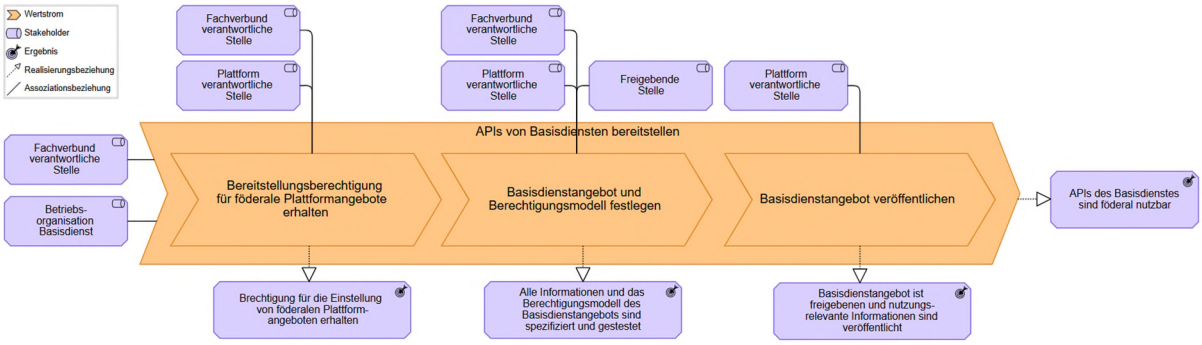


Abbildung 6: Wertstrom „APIs von Basisdiensten bereitstellen“

Tabelle 4: Steckbrief Wertstrom "APIs von Basisdiensten bereitstellen"

Name	APIs von Basisdiensten bereitstellen
Kurzbeschreibung	Dieser Wertstrom beinhaltet alle Aktivitäten zur Bereitstellung von APIs und damit verbundenen Angeboten (Zentrale Frontendanwendungen auf Basis dieser APIs) von Basisdiensten in einem föderalen Plattform-Ökosystem.
Auslösender Stakeholder	Betriebsorganisation Basisdienst: Ist die für einen Basisdienst verantwortliche Stelle, die alle Entscheidungen in Bezug auf die Entwicklung und den Betrieb eines Basisdienstes trifft. Fachverbundverantwortliche Stelle: Ist die für einen Fachverbund verantwortliche Stelle, die einen Fachverbund steuert, der wiederum mehrere Basisdienste und Standards für einen Kontext (bspw. NOOTS oder Innenverwaltung) bündelt und koordiniert.
Wertversprechen	Das zentrale Wertversprechen besteht darin, dass APIs von Basisdiensten effektiv allen nachnutzenden Stellen bereitgestellt werden können und die verantwortlichen Stellen beim Bereitstellungsprozess entlastet werden. Durch die übergreifende Koordinierung der Bereitstellung wird sichergestellt, dass alle Angebote fachlich aufeinander abgestimmt sind und mit konsistenten Prozessen und Berechtigungslogiken bereitgestellt werden.

Tabelle 5: Taskbeschreibung "APIs von Basisdiensten bereitstellen"

Wertstromschritt	Kurzbeschreibung	Ergebnis	Beteiligte Stakeholder
Bereitstellungsberechtigung für föderale Plattformangebote erhalten	Die Aktivitäten beinhalten die Beantragung, Prüfung und Gewährung von Bereitstellungsberechtigungen von föderalen Plattformangeboten.	Die Betriebsorganisation Basisdienst oder die Fachverbundverantwortliche Stelle darf ihre Angebote föderal bereitstellen.	Fachverbundverantwortliche Stelle, Plattformverantwortliche Stelle
Basisdienstangebot und Berechtigungsmodell festlegen	Die Aktivitäten beinhalten die Abstimmung des Basisdienstangebots mit bestehenden Angeboten anderer Basisdienste, die Festlegung des Berechtigungsmodell auf grobgranularer Zugriffsebene und der Integration des Berechtigungsmodells in die übergreifende Berechtigungsstruktur.	Alle Informationen zum Basisdienstangebot und dem Berechtigungsmodell sind spezifiziert und alle Grundlagen für die Berechtigungsprüfung auf grobgranularer Zugriffsebene sind implementiert und getestet.	Fachverbundverantwortliche Stelle, Plattformverantwortliche Stelle, Freigebende Stelle
Basisdienstangebot veröffentlichen	Die Aktivitäten beinhalten die Veröffentlichung von allen Informationen, die notwendig sind, um die Angebote eines Basisdienstes zu nutzen (bspw. Technische Adressen von Betriebsumgebungen, Zertifikate, Voraussetzung für die Nutzung) sowie die Produktivsetzung von Integrationen des Basisdienstes in übergreifende Strukturen (bspw. Zur Registrierung von API-Consumern und dem allgemeinen Berechtigungsmanagement)	Alle Informationen zur Nutzung des Basisdienstangebots sind bereitgestellt und eine produktive Nutzung des Angebots ist möglich.	Plattformverantwortliche Stelle

3.1.3 Wertstrom „Angebote von Basisdiensten nutzen“

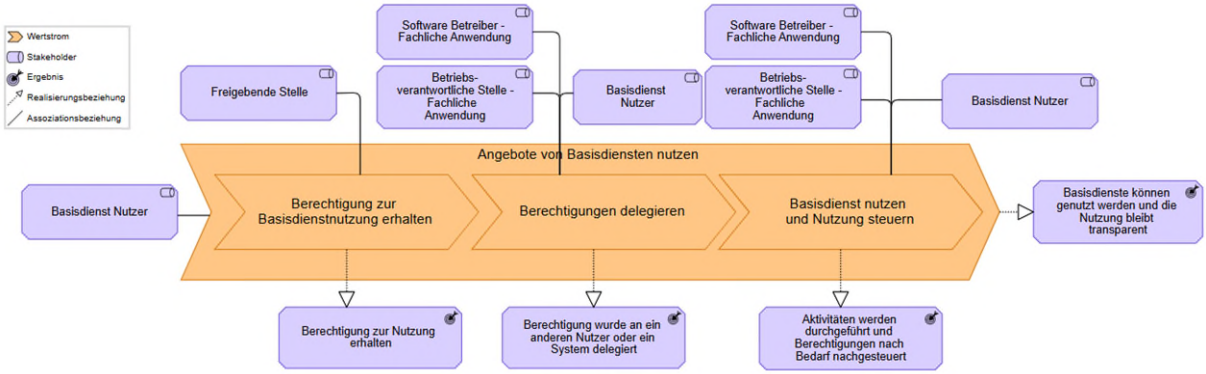


Abbildung 7: Wertstrom „Angebote von Basisdiensten nutzen“

Tabelle 6: Steckbrief Wertstrom "Angebote von Basisdiensten nutzen"

Name	Angebote von Basisdiensten nutzen
Kurzbeschreibung	Der Wertstrom beinhaltet alle Aktivitäten vom Erhalt einer Berechtigung zur Basisdienstnutzung bis hin zur Nutzung des Basisdienstes.
Auslösender Stakeholder	• Basisdienstnutzer: Eine natürliche Person oder Organisation, die einen Basisdienst nutzt. Diese Nutzung kann von regulären Nutzungen als Verwaltungskunde reichen bis hin zur Bearbeitung besonderer oder hoheitlicher Datenbestände.
Wertversprechen	Das zentrale Wertversprechen liegt darin, alle benötigten Basisdienste schnell und einfach zu nutzen. Berechtigungen können transparent an Dritte (fachliche Anwendungen oder andere Basisdienstnutzer) für definierte fachliche Kontexte und Zeiträume delegiert und diese Autorisierungen gesteuert und jederzeit zurückgenommen werden. Für alle beteiligten Stakeholder liegen Berechtigungsinformationen in der Weise vor, dass Sicherheitsinteressen und Datenschutzinteressen im Einklang stehen.

Tabelle 7: Taskbeschreibung "Angebote von Basisdiensten nutzen"

Wert-stromschritt	Kurzbeschreibung	Ergebnis	Beteiligte Stake-holder
Berechtigung zur Basisdienstnut-zung erhalten	Die Aktivitäten beinhalten die initiale Vergabe von Berechtigungsinformationen an Basisdienstnutzer zur Nutzung des Ba-sisdienstes.	Alle benötigten Berechtigungen erhalten	Freigebende Stelle

Berechtigungen delegieren	Die Aktivitäten beinhalten die Autorisierung von dritten Personen und Anwendungen mit der Durchführung von Aktivitäten und der damit verbunden Delegation von Berechtigungen.	Dritte wurden für bestimmte Handlungen autorisiert und die dafür notwendigen Berechtigungsinformationen delegiert	Software Betreiber – Fachliche Anwendung, Betriebsverantwortliche Stelle – Fachliche Anwendung, Basisdienstnutzer
Basisdienst nutzen und Nutzung steuern	Die Aktivitäten beinhalten die Nutzung des Basisdienstes und die Steuerung der Nutzung von autorisierten Dritten.	Basisdienst ist in Nutzung und ggf. vorhandene Aktivitäten von Dritten können transparent gesteuert werden.	Software Betreiber – Fachliche Anwendung, Betriebsverantwortliche Stelle – Fachliche Anwendung, Basisdienstnutzer

3.1.4 Zentrale Beziehungen zwischen den identifizierten Wertströmen

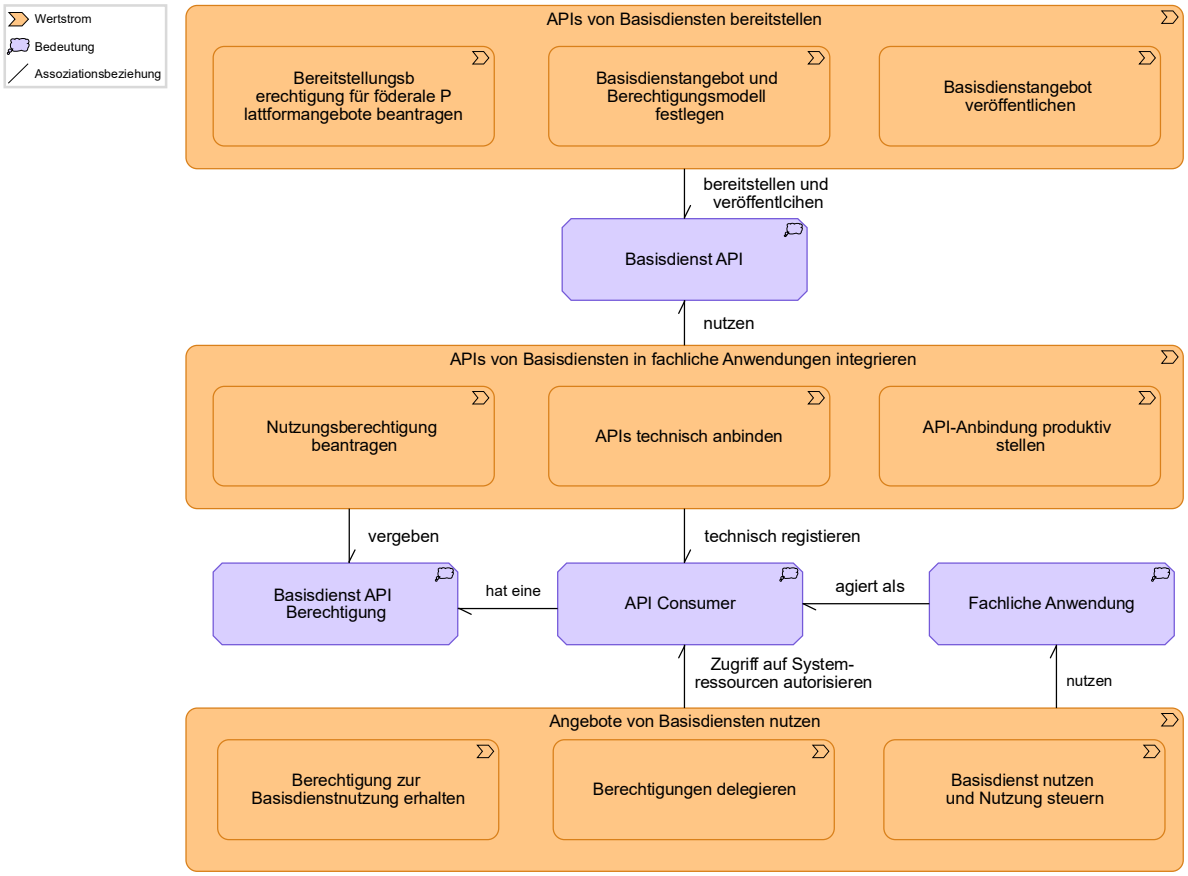


Abbildung 8: Übersicht zentraler gemeinsam genutzter Informationskonzepte

Zwischen den Wertströmen lassen sich zwei zentrale Informationskonzepte (für mehr Informationen siehe Kapitel 3.2) identifizieren, die Wertströme miteinander verknüpfen:

- Basisdienst-API: Die Basisdienst-API wird als Ergebnis zur Bereitstellung von Basisdienst APIs nutzbar und ist wiederum das zentrale Element im Kontext der Integration von fachlichen Anwendungen.
- API-Consumer: Fachliche Anwendungen erhalten in ihrer Rolle als API-Consumer die notwendigen API-Berechtigungen im Rahmen der API-Integration als technische Grundlage, damit fachliche Anwendungen überhaupt auf APIs zugreifen und fachliche Aktionen für ihre Nutzer anbieten können.

3.2 Übersicht der Informationskonzepte

Der vorliegende Abschnitt stellt das Informationskonzept für eine föderale IT-Plattformarchitektur und ihre Beziehungen zueinander dar. Das Verständnis von Informationskonzepten folgt hierbei der Definition eines Informationskonzepts nach BIZBOK:

> *Ein Informationskonzept beschreibt die geschäftlich relevanten Informationen einer Organisation in strukturierter Form. Es identifiziert und definiert die zentralen Geschäftsobjekte, ihre Eigenschaften, ihre Beziehungen zueinander sowie ihre Bedeutung für Geschäftsabläufe und Entscheidungsfindung.*

Um einheitliches Verständnis der Geschäftsarchitektur zu erreichen, werden die identifizierten Informationskonzepte in einer Information Concept Map (siehe Kapitel 3.2.2) in Beziehung zueinander gesetzt. Diese Information Concept Map kann als Vorstufe für detaillierte technische Datenarchitekturen oder Rollen- und Berechtigungsmodelle dienen. Ergänzend zur Information Concept Map werden alle Informationskonzepte und ihre Beziehungen anhand einer Tabelle detailliert definiert und erläutert (siehe Kapitel 3.2.3).

3.2.1 Konzeptionelle Unterscheidung zwischen Berechtigungssteuerung auf API-Ebene und Anwendungsebene bzw. Ressourcenebene

Um die nachfolgenden Informationskonzepte besser einzuordnen, ist ein Verständnis über die zwei zentralen Betrachtungsebenen der Berechtigungssteuerung im Kontext der Gesamtarchitektur von zentraler Bedeutung. Die beiden Ebenen werden in der folgenden Abbildung anhand einer Gebäudemetapher erläutert.

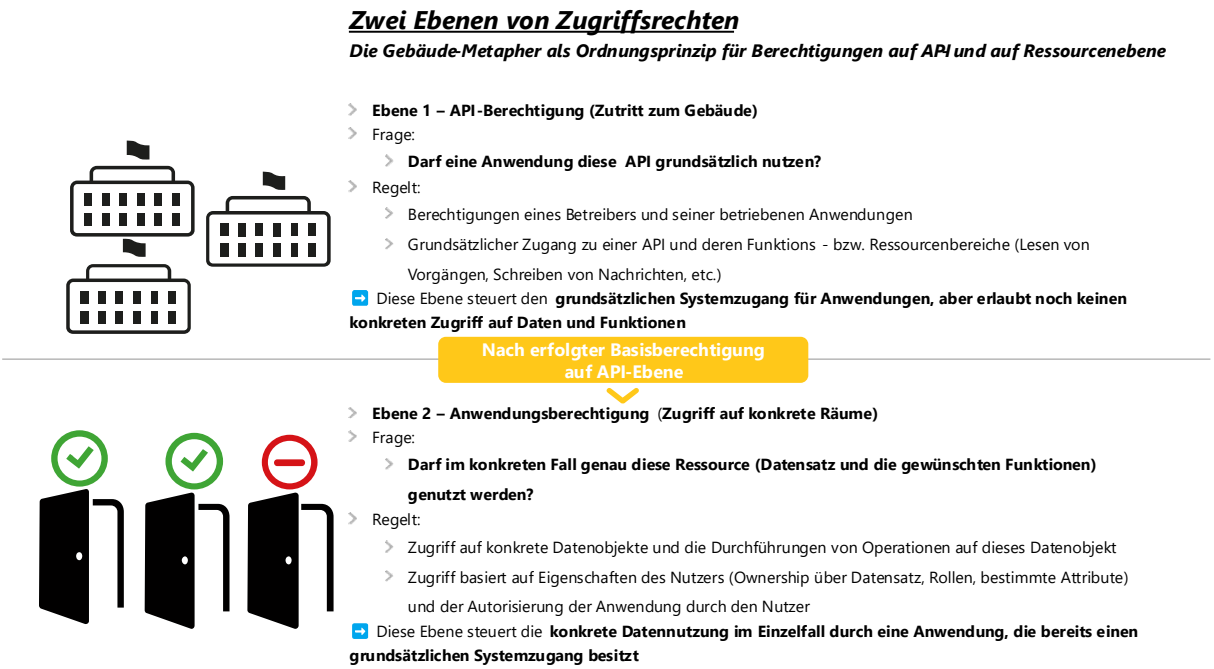


Abbildung 9: Zwei Ebenen der Berechtigungssteuerung

Ein zentraler Aspekt der Gesamtarchitektur ist, dass eine fachliche Anwendung für den Zugriff auf eine API eines Basisdienstes separate Berechtigungen auf zwei Ebenen benötigt:

- Eine grundsätzliche Zugriffsberechtigung auf API-Ebene
- Eine Berechtigung auf konkrete Ressourcen, die diese API bereitstellt

Die grundsätzliche Zugriffsberechtigung wird in der Regel an eine Betriebsverantwortliche Stelle einer fachlichen Anwendung vergeben. Die Berechtigung auf konkrete Ressourcen wird an Basisdienstnutzer vergeben. Ein solcher Basisdienstnutzer kann eine natürliche Person oder eine Organisation sein, der diese Berechtigungen im Rahmen einer Anwendungsnutzung an diese Anwendung für konkrete Aktivitäten und Zeiträume delegiert oder – in Maschine-zu-Maschine-Szenarien (M2M) – eine Software, die dauerhaft mit eigenen Berechtigungen für die Kommunikation ausgestattet wird.

In der Praxis kann es sein, dass eine Betriebsverantwortliche Stelle und ein Basisdienstnutzer identisch sind, bspw. wenn eine Kommune ein eigenes Fachverfahren betreibt und die Kommune eine registerabrufende Stelle ist. In diesem Fall würde das Fachverfahren als API-Consumer von der Kommune in ihren jeweiligen Rollen die beiden Berechtigungen zugewiesen bekommen.

Dass eine solche konzeptionelle Unterscheidung für die Architektur und praktische Umsetzung dennoch elementar ist, kann an zwei Beispielen dargestellt werden:

- Für Onlinedienste, die im Auftrag eines Verwaltungskunden (natürliche Person oder Organisation) agieren, ermöglicht dieses Modell, dass ein Betreiber eines Onlinedienstes nur die Berechtigungen aus der API-Ebene besitzen muss. Die anderen Berechtigungen für den Zugriff auf konkrete Ressourcen (Postfach, Registerdatensatz, Verfahrensakte) kommen in diesem Fall vom Verwaltungskunden und können im Rahmen der Nutzung durch eine temporäre und eingeschränkte Autorisierung vom Verwaltungskunden an den Onlinedienst delegiert werden. Während durch das Fehlen einer solchen Trennung dieser zwei Ebenen oft mit der Fiktion gearbeitet wird, dass die Behörde im Kontext der Onlinedienste agiert, erlaubt dieses Modell flexiblere Bereitstellungsmodelle:
    - Private Dienstleister können als Betreiber von alternativen Onlinediensten wie bei ELSTER üblich auftreten, während die eigentlichen Berechtigungen vom Nutzer kommen und unter dessen Kontrolle liegen.
    - Auch Betreiber von EfA-Diensten können eigene Berechtigungen unabhängig von ihren Auftraggebern erlangen, was eine Skalierung von Nachnutzungsmodellen stark vereinfacht.
- Auch für die Bereitstellung von Fachverfahren ergeben sich große Vereinfachungen und neue Bereitstellungsmodelle. Bereitsteller von SaaS Lösungen (bspw. DVC Cloud Services) und behördenübergreifende Verbünde für den Fachverfahrensbetrieb können ihre gesamte Anwendungsplattform für den Zugriff auf API-Ebene berechtigen und müssen nur noch ihre Services für einzelne Nutzungskontexte oder Betriebsinstanzen um die Anwendungsberechtigungen ihrer Kunden bzw. Endnutzer ergänzen.

Diese Unterscheidung wird in der weiteren Architekturbetrachtung zentral sein, weil Konzepte, Rollen und auch später Systeme bzw. Systemverantwortlichkeiten sich in die genannten Ebenen einordnen werden.

3.2.2 Information Concept Map

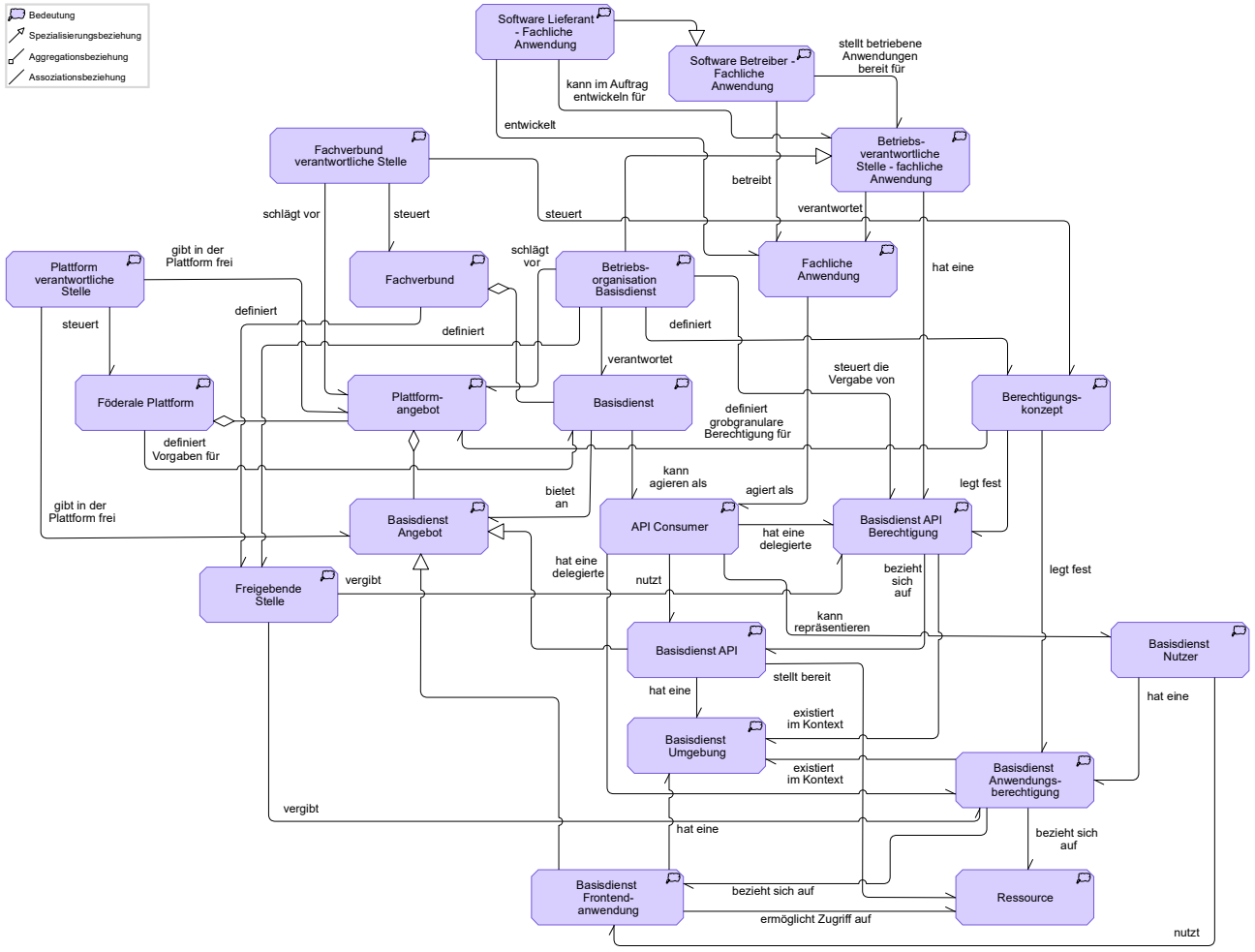


Abbildung 10: Information Concept Map

3.2.3 Beschreibung der Informationskonzepte

Tabelle 8: Beschreibung der Informationskonzepte

Informationskonzept	Kurzbeschreibung	Ausprägungen	Verbundene Informationskonzepte	Praxisbeispiele

API-Consumer	Die Rolle einer Anwendung, wenn diese eine API nutzt bzw. integriert.		• Nutzt eine **Basisdienst-API**
			• Rolle kann durch eine **fachliche Anwendung** wahrgenommen werden, wenn diese die API eines Basisdienstes nutzt
			• Rolle kann durch einen **Basisdienst** wahr-genommen werden, wenn diese die API eines anderen Basisdienstes nutzt
			• Rolle kann durch eine **Basisdienst Frontend-Anwendung** wahrgenommen werden, wenn diese die eigene Basisdienst-API nutzt
			• Rolle kann einen **Basisdienstnutzer** repräsentieren, indem es
			  • durch einen Endnutzer (natürliche Person) für konkrete Handlungen autorisiert wurde
			  • oder eine Organisation in einer M2M Kommunikation vertritt
			• Hat eine von der **Betriebsverantwortlichen Stelle** delegierte **API-Berechtigung**
			• Hat eine vom **Basisdienstnutzer** delegierte **Anwendungsberechtigung**	

Basisdienst	Ein Basisdienst ist ein System, das querschnittliche Funktionen für eine Vielzahl von	Basiskomponente	• Bietet ein **Basisdienstangebot** an. • Wird von einer **Betriebsorganisation Basisdienst** verantwortet	ePayBL, FIT-Connect, DVDV, Registernavigation (NOOTS), BundID,

	Endnutzern (über Endnutzeranwendungen) und / oder Systeme (über APIs) anbietet. Die Entwicklung und der Betrieb eines Basisdienstes erfolgt zentral für eine definierte Nutzergruppe. In der öffentlichen Verwaltung können für die gleiche Funktionalität (bspw. Bezahlung) mehrere Basisdienste (bspw. Payment Dienste) für unterschiedliche Nutzergruppen (bspw. pro Bundesland) existieren.		• Kann auch als **API Consumer** agieren, wenn APIs anderer Basisdienste genutzt werden • Hat eine von der **Betriebsverantwortlichen Stelle** delegierte **API-Berechtigung, wenn er als API-Consumer agiert** • Kann Teil eines **Fachverbund** sein	Nationale Statistikkomponente
Basisdienstangebot	Angebot eines Basisdienst, dass es an andere Systeme oder Endnutzer bereitstellt. Ein Angebot kann als API oder Anwendung für das jeweilige Nutzungsszenario bereitgestellt werden. Beide Bereitstellungsformen können hinsichtlich des Funktionsumfangs identisch sein.		• Wird von einem **Basisdienst** angeboten • Kann eine **Basisdienst-API** sein • Kann eine **Basisdienst-Frontendanwendung** sein • Ist Teil eines **Plattformangebots** • wird von der **Plattformverantwortlichen Stelle** freigegeben	Beispielangebote für APIs: • „Anträge versenden" oder „Anträge empfangen" über die FIT-Connect Submission API, • „Melderegisterdaten abrufen" über die API eines Data Providers in NOOTS Beispielangebote für Frontendanwendungen:

				• „Zustellpunkte verwalten" über FIT-Connect Self-Service-Portal • „Postfachverwaltung und -nutzung" über eine Nutzerkonto Postfachanwendung • „FIM-Leistungen bearbeiten" über FIM Redaktionssystem
Basisdienst Anwendungsberechtigung	Berechtigung auf Level der Anwendungsebene eines Basisdienstes beim Zugriff durch Nutzer. Meint sowohl Coarse-Grained Access Control und Fine-Grained Access Control und regelt den Zugriff auf konkrete Ressourcen des Basisdienstes.		• Wird von einem **Basisdienstnutzer** besessen und kann von einem Nutzer an ein **API-Consumer** delegiert werden, wenn dieser **API-Consumer** den **Basisdienstnutzer** repräsentiert. • Bezieht sich auf den Zugriff auf eine **Basisdienst-Frontendanwendung** (Coarse-Grained) und auf die **Ressource** (Fine-Grained) • Berechtigungslogik wird durch das **Berechtigungskonzept** des Basisdienstes bzw. des Fachverbund definiert • Existiert im Kontext einer **Basisdienstumgebung**	Ein Fachverfahren darf nur auf die eigenen Zustellpunkte in FIT-Connect zugreifen
Basisdienst-API	Eine Basisdienst-API (Appication Programming Interface) ist eine maschinelle Schnittstelle, mit der **ein Basisdienst** seine Funktionen anderen		• Ist Teil eines **Basisdienstangebots** bzw. stellt dieses Angebot **API-Consumern** zur Verfügung • Eine Basisdienst-API hat mehrere **Basisdienstumgebungen**	• FIT-Connect-Submission-API • XBezahldienste-API • ZBP-API (inkl. Statusmonitor-API)

	Anwendungen zur Nutzung zu Verfügung stellt.		• Unterliegt **Basisdienst-API-Berechtigungen**, die den Zugriff auf die API regeln • Stellt API-Consumern **Ressourcen** bereit und ermöglicht Funktions-ausführungen auf diese Ressource (bspw. erstellen, lesen, verändern, löschen)	• SAK-API für Data Provider • SAK-API für Data Consumer
Basisdienst-API-Berechtigung	Berechtigung auf Level der API eines Basisdienstes beim Zugriff durch Systeme. Meint immer ein Coarse-Grained Access Control, der den grundsätzlichen Zugriff auf die API und Bereiche der API regelt.		• Bezieht sich auf den Zugriff auf die API und dortige Funktions- und Ressourcenbereiche (Coarse-Grained) • Berechtigungslogik wird durch das **Berechtigungskonzept** des Basisdienstes bzw. des Fachverbund definiert • API-Berechtigungsvergabe wird von der **Betriebsorganisation Basisdienst** gesteuert • Kann von einer **Betriebsverantwortlichen Stelle** besessen werden und an den von ihr betriebenen API-Consumer delegiert werden. • Existiert im Kontext einer **Basisdienstumgebung**	Ein Onlinedienst darf die Submission-API benutzen
Basisdienstumgebung	Eine Umgebung ist eine Betriebs- bzw. Bereitstellungsvariante der API für spezifische Zwecke wie bspw. Produktivnutzung der API oder Testnutzung bei der Implementierung der API-Anbindung.		• Ist eine Ausprägung einer **Basisdienst-API** • Ist eine Ausprägung einer **Frontendanwendung**	• Test-, Stage- und Produktivumgebung von FIT-Connect. • Demo-Umgebung des PVOG • Schulungs-Umgebung eines Redaktionssystems

Basisdienstnutzer	Ein Basisdienstnutzer ist eine Person oder Organisation, die einen Zugriff auf die Ressourcen und damit verknüpften Funktionen des Basisdienstes hat.	Endnutzer, Accountinhaber	• Hat eine **Basisdienst-Anwendungsberechtigung**, die einen Zugriff auf konkrete Ressourcen und Funktionen ermöglicht. • Kann vom **API-Consumer** repräsentiert werden • Kann **Basisdienst-Frontendanwendung** nutzen	Postfachinhaber beim BundID oder MUK-Postfach, Sender oder Empfänger bei FIT-Connect, Redaktionsmitarbeiter bei FIM
Basisdienst-Frontendanwendung	Eine Basisdienst-Frontendanwendung ist UI für Nutzer des Basisdienstes. Zwei typische Ausprägungen sind vorhanden: Eine Standalone Anwendung für Nutzer, um diese Querschnittsfunktionen bereitzustellen. Ein Beispiel sind die FIM-Editoren für Prozesse oder Datenfelder. Ein Nutzerinterface, um ohne einen externen API-Consumer auf die Basisdienst-API zuzugreifen. Beispiele hierfür sind die FIT-Connect Zustellpunktverwaltung und oder das EfA-Parametrisierungsportal.		• Ist Teil eines **Basisdienstangebots** bzw. stellt dieses Angebot den **Basisdienstnutzern** zur Verfügung • Eine Basisdienst-Frontendanwendung hat mehrere **Basisdienstumgebungen** • Unterliegt **Basisdienst-Anwendungsberechtigungen**, die den Zugriff auf die Anwendungen und Ressourcen regelt • Ermöglicht **Basisdienstnutzern** einen Zugriff auf **Ressourcen** des Basisdienstes • Agiert beim Zugriff auf die eigene Basisdienst-API als **API-Consumer** und nutzt im Sinne des API-First Ansatzes auch die gleichen Zugriffsmechanismen wie externe Anwendungen.	• FIT-Connect-Self-Service-Portal • BundID-Self-Service-Portal • AMS/DAMAS (für NOOTS) • FIM-Portal • PVOG-Such-Client
Berechtigungskonzept	Legt die Strukturen und Berechtigungslogik und die dafür notwendigen Voraussetzungen für den Zugriff auf Ressourcen und Funktionen		• definiert die Berechtigungs-steuerung für das **Plattformangebot und** damit die darin enthaltenen **Basisdienstangebote** • Legt die Berechtigungslogik für die **Basisdienst-API-Berechtigung** fest	

	von Basisdienstangeboten fest. Der Umfang eines Berechtigungskonzept kann einen Basisdienst oder einen kompletten Fachverbund umfassen. Ein Berechtigungskonzept kann in zwei Ebenen unterschieden werden (siehe auch 3.2.1): Coarse-Grained Access Control (Grobgranulare Berechtigungs-steuerung): Coarse-Grained Access Control ist eine Zugriffskontrolle, bei der Berechtigungen auf einer allgemeinen oder aggregierten Ebene für größere Ressourcengruppen oder ganze Systeme definiert werden. Fine-Grained Access Control (Feingranulare Berechtigungs-steuerung): Fine-Grained Access Control ist eine Zugriffskontrolle, bei der Berechtigungen auf einer detaillierten Ebene einzelner Ressourcen, Attribute oder Operationen definiert und durchgesetzt werden.		• Legt die Berechtigungslogik für die **Basisdienst-Anwendungsberechtigung** fest • Wird von der **Betriebsorganisation Basisdienst** definiert • Wird vom **Fachverbundverantwortliche Stelle** gesteuert	

Betriebsorganisation Basisdienst	Ist die Betriebsverantwortliche Stelle für den Basisdienst. Die Betriebsorganisation Basisdienst kann alle Entscheidungen in Bezug auf den Betrieb, die Weiterentwicklung und sonstige Ausgestaltung des Basisdienstes treffen. Eine solche Betriebsorganisation kann Teil einer größeren Governancestruktur sein, in der zentrale Entscheidungen in Bezug auf den Basisdienst getroffen werden.		• Verantwortet einen **Basisdienst** • Ist eine besondere Ausprägung einer **Betriebsverantwortlichen Stelle** • Schlägt eine **Plattformangebot** für das **Basisdienstangebot** vor • Definiert das **Berechtigungskonzept** des **Basisdienstes** • Steuert die Vergabe der **Basisdienst-API-Berechtigung** an betriebsver**antwortliche Stellen**	FITKO bei FIT-Connect, DVDV, BVA für IDA, BMDS bei BundID
Betriebs-verantwortliche Stelle – Fachliche Anwendung	Eine Betriebsverantwortliche Stelle verantwortet den Betrieb einer Softwarelösung und ist damit für alle Aspekte der Datenverarbeitung verantwortlich. Dazu gehört auch die Einhaltung von Nutzungsbedingungen für Einbindung von Basisdienst-APIs.		• Lässt vom Software Betreiber eine fachliche Anwendung betreiben • Kann mit der Entwicklung einer fachlichen Anwendung den Software Lieferant beauftragen • verantwortet Fachliche Anwendungen • hat Basisdienst-API-Berechtigungen • kann auch die Betriebsorganisation eines Basisdienstes sein	Themenfeldführende Behörde, die einen EfA-Onlinedienst entwickeln lässt/lassen Leitende Organisation eines Entwicklungs-verbunds von Redaktionssystemen
Fachliche Anwendung	Anwendung zur Unterstützung fachlicher Aufgaben oder Prozesse.	Onlinedienst, Fachverfahren, Unternehmens-anwendung, System	• Agiert als **API-Consumer** bei der Nutzung von **Basisdienst-APIs** • Wird von einer **betriebsverantwortlichen Stelle** verantwortet • wird von einem **Software Lieferant** entwickelt	Bauportal, Online Anwohnerparkausweis Antragsdienst, Meldefachverfahren, ERP-System eines Unternehmens

Fachverbund	Ein Fachverbund ist ein logisch abgegrenzter Verbund aus fachlichen, organisatorischen und technischen Komponenten, die auf Grundlage gemeinsamer fachlicher Ziele, Regeln und Standards zusammenwirken, um eine definierte fachliche Aufgabe oder einen fachlichen Anwendungsbereich zu unterstützen. Ein Fachverbund kann bspw. Fachverfahren, Register, Basisdienste, technische Infrastrukturen und beteiligte Organisationen umfassen. Die Komponenten bleiben dabei organisatorisch und technisch eigenständig, sind jedoch durch gemeinsame Schnittstellen, Vereinbarungen und Governance-Mechanismen miteinander verbunden.		• Ein Fachverbund beinhaltet eine oder mehrere **Basisdienste**, wobei die jeweiligen Basisdienste nicht zwingend einem Fachverbund zugeordnet sein müssen. • Ein Fachverbund wird von einer **Fachverbundverantwortlichen Stelle** geleitet.	Beispiele für Fachverbünde sind das NOOTS mit den dortigen NOOTS Komponenten und Standards oder der Fachverbund der Innenverwaltung mit u.a. XInneres, DVDV, Governikus und der V-PKI.
Fachverbundverantwortliche Stelle	Die Fachverbundverantwortliche Stelle ist verantwortlich für einen Fachverbund und regelt das dortige Zusammenspiel von Systemen und Organisation, indem es Standard sowie organisatorische und		• Steuert einen **Fachverbund** • Schlägt **Plattformangebote** vor • Steuert für verantwortete **Basisdienste** die **Berechtigungskonzepte** von **Plattformangeboten**	FITKO als fachlich koordinierende Stelle für NOOTS gemäß NOOTS Staatsvertrag

	technische Vorgaben für alle Teilnehmer des Verbunds verabschiedet. Dabei regelt sie auch das dortige Angebot von Basisdiensten und auch die Zugangsregeln und Berechtigungsmodelle über alle Basisdienste hinweg. Eine solche Stelle kann Teil einer größeren Governancestruktur sein, in der zentrale Entscheidungen in Bezug auf den Fachverbund getroffen werden.			
Föderale Plattform	Gemeinsames föderales Angebot an abgestimmten Basisdiensten (sowie weiteren Angeboten außerhalb des Scope dieses Konzepts) von Bund, Ländern und Kommunen.		• Wird von der **Plattformverantwortlichen Stelle** gesamthaft gesteuert / koordiniert. • Enthält bestimmte **Plattformangebote** • Definiert Vorgaben (bspw. Qualitäts-, Betriebs-, Sicherheitsvorgaben) für **Basisdienste**, die Plattformangebote bereitstellen	
Freigebende Stelle	Eine Freigebende Stelle ist eine Stelle, die damit betraut wurde Berechtigungen für die Nutzung von Basisdienstangeboten zu erteilen, wenn für die Vergabe eine mensch-		• Konkrete Stelle und die Verantwortlichkeiten werden definiert durch den **Betriebsorganisation Basisdienstes** • • Vergibt **Basisdienst-API Berechtigungen**	

	licher Entscheidungsspielraum besteht.		• Vergibt **Basisdienst-Anwendungsberechtigungen**	
Plattformangebot	Ein Plattformangebot definiert ein Angebotsklasse eines Basisdienstangebots für die föderale Plattform. Eines solche Angebotsklasse kann mehrere funktional identische Basisdienstangebote gruppieren (bspw. Payment, Identityprovisionierung oder Registerabruf). Eine Angebotsklasse kann auf einen gemeinsamen Standard referenzieren, der für die darin beinhalteten Basisdienstangebote bspw. API-Schnittstellendefinition oder auch die Berechtigungslogik einheitlich definiert. Eine Klasse kann auch auf einen Fachverbund referenzieren, der Basisdienst Angebote dieses Fachverbunds gruppiert und ggf. dafür übergreifend Berechtigungen definiert.		• Bündelt und strukturiert **Basisdienstangebote** • Ist Teil der **föderalen Plattform** • Wird von der **Betriebsorganisation Basisdienst** vorgeschlagen • Wird von der **Fachverbundverantwortlichen Stelle** vorgeschlagen • Wird von der **Plattformverantwortlichen Stelle** definiert und freigegeben • bekommt durch das **Berechtigungskonzept** Berechtigungen definiert	Payment, Melderegisterabruf, Verfahrensnachrichtenaustausch
Plattformverantwortliche Stelle	Die plattformverantwortliche Stelle ist verantwortlich für den Zugang zur gemeinsamen Plattforminfrastruktur für Anbieter von Basisdienst-		• Definiert, standardisiert und gibt **Plattformangebote** für die Plattform frei. • Gibt einzelne **Basisdienstangebote** für die Plattform frei	

	angeboten und für API Consumer, die über API die Basisdienste in ihre Lösungen integrieren. Für diesen Zweck steuert die Plattformverantwortliche Stelle übergreifende Standards bspw. im Bereich von API-Autorisierung, Berechtigungsverwaltung und basisdienstrelevanter Identitäten. Zudem verantwortet diese Stelle alle relevanten zentralen Infrastrukturen für die nahtlose Nutzung und Integration von Basisdiensten im föderalen Ökosystem.		• steuert die **Plattformangebote** der **föderalen Plattform**	
Ressource	Eine Ressource ist ein durch einen Resource Server (API-Provider) kontrolliertes, eindeutig identifizierbares Zielobjekt oder eine Zieloperation, dessen Zugriff durch Autorisierungs-mechanismen gesteuert wird und das über eine Schnittstelle adressierbar ist.		• Wird durch **Basisdienst-API** bereitgestellt und erlaubt Operationen auf der Ressource gemäß den jeweiligen Berechtigungen • **Basisdienst Anwendungs-berechtigungen** haben einen Bezug auf konkrete Ressourcen (Berechtigung von Nutzer A auf Dokument X zuzugreifen) • **Basisdienst-Frontendanwendungen** ermöglichen den **Basisdienstnutzern** über die **Basisdienst-API** einen Zugriff auf Ressourcen	Antrag, Verfahrensvorgang, Registereintrag, Postfach, Zahlungsvorgang, FIT-Connect Zustellpunkt, FIM / XZuFi Leistungsbeschreibung, FIM-Datenfeld, FIM-Prozess, DVDV Serviceeintrag.
Software Betreiber – Fachliche Anwendung	Der Software Betreiber einer fachlichen Anwendung ist eine Stelle, die damit betraut	Betriebsorganisation der betriebsverantwortlichen	• Betreibt für die Betriebsverantwortliche Stelle eine Fachliche Anwendung	Öffentliche und private Betreiber wie bspw. ITZBund, HZD, AKDB,

	ist im Auftrag einer betriebsverantwortlichen Stelle eine fachliche Anwendung zu betreiben.	Stelle (Abteilung oder ausgelagerter interner IT-Dienstleister), SaaS Anbieter		KommOne, ekom21, Dataport, IONOS
Software Lieferant – Fachliche Anwendung	Ein Software Lieferant entwickelt eine fachliche Anwendung entweder als Standardsoftware oder als Individualsoftware im Auftrag Dritter oder als interne Leistung gegenüber anderen Organisationseinheiten oder der jeweiligen Gesamtorganisation.	Entwickler von Individuallösungen, Entwickler von Standardlösungen,	• Entwickelt eine **fachliche Anwendung** eigenständig oder im Auftrag für Dritte • Kann eine **fachliche Anwendun**g im Auftrag einer **Betriebsverantwortlichen Stelle** entwickeln • Kann als **Softwarebetreiber – Fachliche Anwendung** auftreten	Öffentliche und private Software Entwickler wie bspw. AKDB, Prosoz, FJD, Publicplan

3.3 Ableitung der benötigten strategischen Fähigkeiten

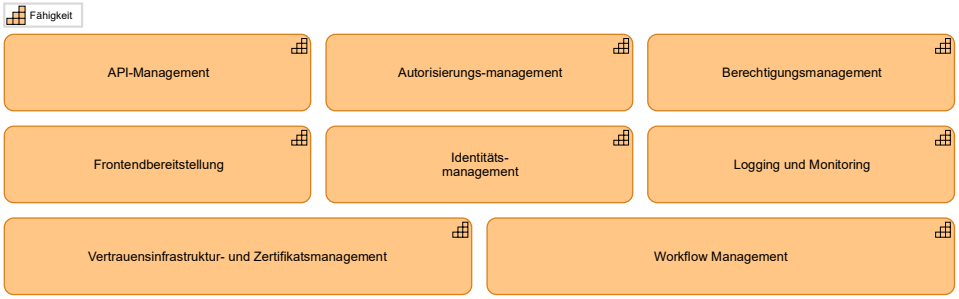


Abbildung 11: Strategische Fähigkeiten

Legende der Abkürzungen:

Wertströme: W1 = APIs von Basisdiensten integrieren | W2 = APIs von Basisdiensten bereitstelle | W3 = Angebote von Basisdiensten nutzen

Tabelle 9: Beschreibung der strategischen Fähigkeiten

Fähigkeit	Kategorie	Definition	Wertströme
API-Management	Kern	Die Fähigkeit, APIs als verwaltete Ressourcen zu behandeln: von der Registrierung und Beschreibung über die Zugangskontrolle bis zur laufenden Prüfung eingehender Anfragen.	W1, W2
Verwaltung von API-Consumern	Kern	Die Fähigkeit, Softwareanwendungen und deren Betreiber als API-Konsumenten zu registrieren, zu identifizieren und zu verwalten – einschließlich der Pflege von Metadaten, Zugangsdaten und Berechtigungszuordnungen.	W1
Verwaltung von API-Katalogen	Kern	Die Fähigkeit, das Angebot verfügbarer APIs strukturiert zu erfassen, zu beschreiben und bereitzustellen – sodass Konsumenten APIs auffinden, verstehen und ihre Integration vorbereiten können.	W1, W2
Prüfung von API-Anfragen	Kern	Die Fähigkeit, eingehende API-Anfragen zur Laufzeit gegen die dem API-Client übertragenen Rechte zu prüfen – insbesondere hinsichtlich Authentizität und Gültigkeit des Tokens sowie der Übereinstimmung der enthaltenen Scopes und Policies mit der konkreten Anfrage.	W1, W3
API-Lifecycle-Management	Kern	Die Fähigkeit, APIs über ihren Lebenszyklus zu verwalten – insbesondere die Steuerung von Versionierung und Deprecation von API-Angeboten sowie die koordinierte Migration betroffener API-Consumer auf neue Versionen.	W1, W2
Workflow Management	Kern	Die Fähigkeit, strukturierte Abläufe innerhalb der föderalen Infrastruktur zu definieren, zu steuern und auszuführen – insbesondere mehrstufige Prozesse, die mehrere Akteure, Systeme oder Entscheidungsschritte umfassen, sowie die attributbasierte Freigabe durch zuständige Stellen.	W1, W2, W3
Workflowdefinition	Kern	Die Fähigkeit, Arbeitsabläufe und Prozessschritte zu modellieren und zu konfigurieren – also festzulegen, welche Schritte in welcher Reihenfolge, unter welchen Bedingungen und durch welche Akteure ausgeführt werden.	W1, W2, W3
Workflow-Ausführung	Kern	Die Fähigkeit, definierte Workflows zur Laufzeit zu instanziieren, zu steuern und zu überwachen – einschließlich der Zuweisung von Aufgaben an beteiligte Akteure und der Verwaltung des Ausführungsstatus.	W1, W2, W3

Aufgabenverwaltung	Kern	Die Fähigkeit, Aufgaben, die im Rahmen von Workflows an beteiligte Akteure zugewiesen werden, zu verwalten – einschließlich der Priorisierung, Weiterleitung und Nachverfolgung offener Aufgaben sowie der Benachrichtigung zuständiger Stellen.	W1, W2, W3
Autorisierungsmanagement	Kern	Die Fähigkeit, Nutzern und Anwendungen auf Basis von Delegationen, Identitäten und geltenden Berechtigungsregeln Zugriffsrechte zu übertragen – in Form von Nachweisen, die die gewährten Rechte für einen definierten Zeitraum und Kontext verbindlich repräsentieren.	W1, W2, W3
Nutzerautorisierung	Kern	Die Fähigkeit, einem Nutzer Zugriffsrechte auf Basisdienste zu übertragen – auf Basis einer expliziten Autorisierung durch einen anderen Nutzer, eine Organisation oder eine bevollmächtigte Stelle. Dies umfasst auch die organisationale Delegation, bei der eine Organisation einem externen Dienstleister oder organisationsfremden Nutzer Rechte überträgt.	W1, W3
Anwendungsautorisierung	Kern	Die Fähigkeit, einer Anwendung Zugriffsrechte auf Basisdienste zu übertragen – mit oder ohne vorgelagerte Nutzerautorisierung – auf Basis der registrierten Eigenschaften der Anwendung und der geltenden Berechtigungsregeln.	W1, W3
Identitätsmanagement	Kern	Die Fähigkeit, digitale Identitäten von Personen und Organisationen innerhalb der föderalen Infrastruktur zu verwalten – von der initialen Registrierung über die laufende Pflege von Identitätsdaten bis zur Authentifizierung.	W1, W2, W3
Account-Management	Kern	Die Fähigkeit, Nutzerkonten und Organisationskonten innerhalb der Plattform anzulegen, zu verwalten und zu deaktivieren – einschließlich der Abbildung organisatorischer Hierarchien mit Unterbereichen, Mitarbeitern und externen Zugehörigkeiten sowie der Zuordnung von Rollen.	W1, W2, W3
Nutzerauthentifizierung	Kern	Die Fähigkeit, die Identität eines Nutzers zum Zeitpunkt des Zugriffs zu verifizieren – entweder durch eigene Mechanismen oder durch Föderierung mit externen Identitätsnachweisgebern.	W1, W2, W3
Verwaltung von Identitätsattributen	Kern	Die Fähigkeit, strukturierte Attributinformationen zu Nutzern und Organisationen zu pflegen – z.B. Behördenzugehörigkeit, Rolle oder Zertifizierungsnachweise – die als Grundlage für Berechtigungsentscheidungen dienen.	W1, W2, W3

Föderierungsmanagement	Kern	Die Fähigkeit, Vertrauensbeziehungen zu externen Identitätsnachweisgebern zu konfigurieren und zu verwalten – einschließlich der Festlegung, welchen externen Quellen vertraut wird und unter welchen Bedingungen deren Nachweise akzeptiert werden.	W1, W2
Prüfung von Nutzerzugriffen	Kern	Die Fähigkeit, zu prüfen und zu verwalten, ob und in welchem Umfang ein Nutzer einem Client die Berechtigung erteilt hat, in seinem Namen auf Basisdienste zuzugreifen – einschließlich der Verwaltung erteilter Zustimmungen und deren Widerruf.	W3
Integration externer Attributquellen	Kern	Die Fähigkeit, externe Attribute Authorities als vertrauenswürdige Datenquellen einzubinden und die von ihnen ausgestellten Attributnachweise – etwa fachspezifische Zertifizierungen oder Registrierungsnachweise – in die Infrastruktur zu übernehmen und bereitzustellen.	W1, W2
Berechtigungsmanagement	Kern	Die Fähigkeit, die Regeln und Informationen zu verwalten, auf deren Basis Autorisierungsentscheidungen getroffen werden – also das normative Fundament der gesamten Zugriffskontrolle.	W1, W2, W3
Verwaltung von Berechtigungsregeln	Kern	Die Fähigkeit, Zugriffsregeln zu definieren, zu pflegen und bereitzustellen – einschließlich der Festlegung von Bedingungen, Ausnahmen und den damit verbundenen Scopes.	W1, W2, W3
Verwaltung von Berechtigungsinformationen	Kern	Die Fähigkeit, die Attributinformationen zu verwalten, die zur Auswertung von Berechtigungsregeln herangezogen werden – also die faktische Datenbasis, gegen die Regeln zur Laufzeit geprüft werden.	W1, W2, W3
Logging und Monitoring	Kern	Die Fähigkeit, alle relevanten Ereignisse und Systemaktivitäten der föderalen Infrastruktur zu erfassen, zu speichern und bereitzustellen – sowohl für die laufende Betriebsüberwachung als auch für nachgelagerte Prüfungen und Auditierungen.	W1, W2, W3
Ereignisprotokollierung	Kern	Die Fähigkeit, sicherheits- und betriebsrelevante Ereignisse der Infrastruktur manipulationssicher zu erfassen und zu speichern – als revisionssicherer Prüfpfad für Auditoren, Betreiber und Prüfinstanzen.	W1, W2, W3
Anomalieerkennung	Kern	Die Fähigkeit, ungewöhnliche Muster und potenzielle Sicherheitsvorfälle in der Infrastruktur zu erkennen und betroffene Stellen zu benachrichtigen – als Grundlage für eine aktive Reaktion auf Angriffe oder Regelverstöße.	W1, W2, W3

Betriebsüberwachung	Kern	Die Fähigkeit, den laufenden Systemzustand und die Verfügbarkeit der Infrastrukturkomponenten zu überwachen – als Grundlage für einen stabilen und unterbrechungsfreien Betrieb.	W1, W2, W3
Berichterstattung & Auswertung	Kern	Die Fähigkeit, erfasste Ereignis- und Betriebsdaten auszuwerten und strukturiert bereitzustellen – etwa für Compliance-Nachweise, Nutzungsstatistiken oder Sicherheitsanalysen.	W1, W2, W3
Vertrauensinfrastruktur- und Zertifikatsmanagement	Kern	Die Fähigkeit, die kryptographische Vertrauensbasis der gesamten Infrastruktur zu verwalten – einschließlich der Ausstellung von Vertrauensnachweisen, der Pflege von Vertrauensbeziehungen und der sicheren Verwaltung kryptographischer Schlüssel.	W1, W2, W3
Ausstellung von Vertrauensnachweisen	Kern	Die Fähigkeit, autoritative Nachweise auszustellen, die die Vertrauenswürdigkeit von Akteuren und Systemkomponenten gegenüber der Infrastruktur belegen – als Grundlage für sichere Kommunikation und Identifikation.	W1, W2
Verwaltung von Vertrauensbeziehungen	Kern	Die Fähigkeit, Vertrauensbeziehungen zu externen und internen Vertrauensankern zu verwalten – einschließlich der Festlegung, welchen Stellen die Infrastruktur vertraut und unter welchen Bedingungen deren Nachweise anerkannt werden.	W1, W2
Schlüsselverwaltung	Kern	Die Fähigkeit, kryptographische Schlüssel der Infrastruktur sicher zu verwalten – einschließlich ihrer Erneuerung und ihres Entzugs.	W1, W2, W3
Frontendbereitstellung	Unterstützend	Die Fähigkeit, Nutzern digitale Oberflächen bereitzustellen, über die sie mit den Funktionen und Diensten einer Plattform interagieren können.	W1, W2, W3
Frontend-Integration	Unterstützend	Die Fähigkeit, Oberflächen und Funktionen verschiedener unabhängig betriebener Systeme für Nutzer zu einem kohärenten Nutzererlebnis zusammenzuführen – sodass Systemgrenzen aus Nutzerperspektive nicht wahrnehmbar sind.	W1, W2, W3
Content Management	Unterstützend	Die Fähigkeit, die Inhalte der zentralen Plattform als Zugang und Informationshub strukturiert zu pflegen – einschließlich der Beschreibung von Angeboten, Nutzungsbedingungen und Prozessanleitungen.	W1, W2, W3

4 IT-Architektur

Die IT-Architektur der föderalen API-Autorisierungsinfrastruktur beschreibt die technische Umsetzung der in der Geschäftsarchitektur identifizierten Fähigkeiten und Wertströme. Das Kapitel gliedert sich in drei Abschnitte:

- Zunächst werden die Rahmenbedingungen dargelegt, die die Konzeption der Architektur leiten – darunter verbindliche Architekturvorgaben, relevante Industriestandards sowie die getroffenen Architekturentscheidungen.
- Daran anschließend beschreibt die Übersicht der Systemlandschaft die beteiligten Systeme und ihre Rolle in der Gesamtarchitektur.
- Die technische Informationsarchitektur legt schließlich dar, welche Datenobjekte im System existieren, wie die Informationsverantwortung verteilt ist und wie Informationen zwischen den Systemkomponenten fließen.

4.1 Rahmenbedingungen für die Konzeption der IT-Architektur

Die Konzeption der IT-Architektur erfolgt auf Basis verbindlicher Vorgaben sowie anerkannter Industriestandards und Best Practices. Dieses Kapitel beschreibt die wesentlichen Grundlagen, die die Architektur maßgeblich prägen: die normativen Vorgaben des Projekts und der öffentlichen Verwaltung, die betrachteten Standards und Protokolle im Bereich OAuth und API-Sicherheit sowie die im Projektverlauf getroffenen Architekturentscheidungen.

4.1.1 Architekturvorgaben

Die Architektur der föderalen API-Autorisierungsinfrastruktur basiert auf folgenden verbindlichen Vorgaben, die bei der Konzeption berücksichtigt wurden:

- Die vom Projekt definierten Vorgaben zum Schutzbedarf hoch und sehr hoch (https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/blob/85ede9148efc022fe00f8197cce968f7aca9da57/Vorgaben/Schutzbedarf_hoch_und_sehr_hoch.md)
- BSI-Grundschutz in der aktuellen Fassung
- Föderale IT-Architekturrichtlinien in aktueller Fassung

4.1.2 Betrachtete Best-Practices und Industriestandards

Bei der Konzeption der Architektur wurden etablierte Best Practices zur IT-Sicherheit und zum Architekturdesign im Kontext von API-Autorisierung und föderalen Infrastrukturen berücksichtigt. Dazu zählen insbesondere Sicherheitsempfehlungen und Architekturmuster aus dem OAuth-Ökosystem, Umsetzungserfahrungen führender OAuth- und OpenID-Connect-Software-Provider sowie Praktiken aus dem Kontext Open Banking – insbesondere hinsichtlich der automatischen Registrierung von API-Clients in verteilten Softwarelandschaften (Open Banking Implementation Entity, OBIE).

Als technische Grundlage wurden folgende Protokolle und Sicherheitsprofile herangezogen: die aktuellen Spezifikationen der OAuth 2.0-Familie einschließlich des Entwurfs zu OAuth 2.1, die OpenID-Connect-Familie, SCIM, AuthZEN (OpenID Foundation) und das Shared Signals Framework (SSF). Darüber hinaus wurden alle FAPI-2.0-Spezifikationen (Financial-grade API) sowie die formale Sicherheitsanalyse von FAPI 2.0 betrachtet. FAPI 2.0 bildet das primäre Sicherheitsprofil der Zielarchitektur, da es für Hochsicherheitsszenarien entwickelt wurde und explizite Vorgaben zu Client-Authentifizierung, Sender-Constraining und Token-Binding macht.

4.1.3 Dokumentation von Architekturentscheidungen

Die nachfolgende Tabelle verzeichnet alle im Rahmen des Projekts dokumentierten Architekturentscheidungen (Architecture Decision Records, ADRs). ADRs beschreiben wesentliche Entscheidungen zur Systemarchitektur, die geprüften Alternativen, die Entscheidungstreiber sowie die Konsequenzen der getroffenen Wahl. Die ADRs sind im Projektrepository auf OpenCoDE versioniert und verwaltet: https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/tree/b222b7ed46679425f52e78fe406b04649a16904f/Architekturentscheidungen.

Tabelle 10: Architekturentscheidungen

ADR-Nr.	Titel	Kurzbeschreibung des Entscheidungsgegenstandes
ADR-001	Zu verwendender Autorisierungsstandard	Klärt, welcher Autorisierungsstandard als Grundlage der föderalen API-Autorisierungsinfrastruktur verwendet wird. Betrachtete Optionen waren u.a. SAML, SPIFFE/SPIRE sowie verschiedene OAuth-2.0-Profile. Entschieden wurde für OAuth 2.0 mit einem qualifizierten Profil auf Basis von FAPI 2.0, da dieser Standard weit verbreitet, formal analysiert und durch zertifizierte Implementierungen breit unterstützt wird.
ADR-002	Client-Authentifizierung	Klärt, welche Methode für die Authentifizierung von API-Clients gegenüber Authorization Servern verwendet wird. FAPI 2.0 lässt nur mTLS und `private_key_jwt` als ausreichend sicher zu. Entschieden wurde für private_key_jwt, da es vollständig auf Applikationsebene implementierbar ist und aufwändige Abstimmungen mit zentralen Netzwerkbetriebsabteilungen vermeidet.
ADR-003	Sender-Constraining	Klärt, durch welchen Mechanismus sichergestellt wird, dass ein ausgestelltes Access Token nur vom rechtmäßigen Empfänger verwendet werden kann. Betrachtete Optionen waren mTLS-bound Tokens, DPoP und klassische Bearer Tokens. Entschieden wurde für DPoP, da es keine zusätzliche PKI-Infrastruktur erfordert, breite Clientlandschaften unterstützt und sich in das gewählte OAuth/FAPI-Ökosystem einfügt.
ADR-004	Absicherung und Vertrauenswürdigkeit von Public Keys	Klärt, wie die Integrität und Vertrauenswürdigkeit der Public Keys von API-Clients sichergestellt wird. Entschieden wurde, den Public Key in der Software Statement Assertion zu verankern, da die kryptographische Signatur der SSA-Manipulation ausschließt und Key-Änderungen dadurch auditierbar und revisionssicher sind, ohne zusätzliche PKI-Systeme zu benötigen.
ADR-005	Prüfung der API-Autorisierung – Zentraler vs. Dezentraler OAuth-Server	Klärt, ob die API-Autorisierungsprüfung zentral oder dezentral erfolgen soll. Da unterschiedliche Plattformangebote und Basisdienste sehr unterschiedliche Anforderungen haben, wurde entschieden, die Entscheidung dem jeweiligen Fachverbund zu überlassen, der die Rahmenbedingungen seiner Domäne am besten beurteilen kann.
ADR-006	Absicherung der Schnittstellen zwischen Komponenten der zentralen Infrastruktur	Klärt, wie die Kommunikation zwischen den Komponenten der zentralen Infrastruktur abgesichert wird. Da es sich um rein maschinelle Kommunikation innerhalb einer stabilen Infrastruktur handelt, wurde mTLS mit einer begrenzten internen PKI gewählt, das robuste, auditierbare Sicherheit ohne Token-Verwaltungsaufwand bietet.

ADR-007	Absicherung von Schnittstellen zwischen einheitlich bereitgestellten Komponenten der dezentralen Infrastruktur	Klärt, wie Schnittstellen zwischen einheitlich bereitgestellten dezentralen Komponenten wie PDP und PIP abgesichert werden. Entschieden wurde, denselben Standard wie für externe API-Clients zu verwenden, um ein konsistentes Sicherheitsmodell zu gewährleisten und keinen zweiten Sicherheitsstack aufzubauen.
ADR-008	Berechtigungsarchitektur	Klärt, wie die Berechtigungsarchitektur zwischen zentraler Plattform- und dezentraler Fachverbundebene aufgeteilt wird. Entschieden wurde für eine zweistufige Architektur: Grobgranulare Zugangsentscheidungen werden zentral auf Plattformebene gesteuert; feingranulare fachliche Entscheidungen verbleiben in der Verantwortung der Fachverbünde.
ADR-009	Berechtigungsmodell für zentrale Regeladministration	Klärt, welches Berechtigungsmodell für die Verwaltung zentraler Berechtigungsregeln verwendet wird. Entschieden wurde für ein ReBAC-Modell, da es die organisatorischen Beziehungsstrukturen der föderalen Verwaltung abbildet und kontrollierte Delegation entlang der Betreiberstrukturen ermöglicht.
ADR-010	Berechtigungsmodell für Plattformberechtigungen	Klärt, wie Plattformberechtigungen für API-Clients modelliert und verteilt werden. Entschieden wurde für ein regelbasiertes, attributgesteuertes Modell mit verteilter PDP-Nutzung: Attribute werden zentral gepflegt und signiert, Berechtigungsentscheidungen können dezentral getroffen werden ohne Laufzeitabhängigkeit von zentralen Diensten.
ADR-011	Externalisierung von Berechtigungen und Policy Decision Points	Klärt, ob und in welchem Umfang Berechtigungsentscheidungen an einen externen Policy Decision Point ausgelagert werden sollen. Entschieden wurde, die Nutzung des einheitlich bereitgestellten PDP verbindlich vorzuschreiben, sofern das System technisch zur Externalisierung fähig ist; andernfalls werden keine Vorgaben gemacht.
ADR-012	Prüfung der Gültigkeit von grobgranularen Berechtigungen	Klärt, wie die Gültigkeit grobgranularer Berechtigungen zur Laufzeit geprüft wird. Entschieden wurde für eine Kombination aus signiertem Token und PDP: Grobgranulare Berechtigungen werden in Tokens transportiert und lokal prüfbar gemacht, während der PDP ergänzende Entscheidungen trifft, ohne als ständig benötigte Zentralinstanz zu wirken.
ADR-013	Prüfung der Gültigkeit von feingranularen Berechtigungen	Klärt, wie die Gültigkeit feingranularer Berechtigungen innerhalb eines Fachverbunds geprüft wird. Da feingranulare Berechtigungen fachliche Entscheidungsgegenstände sind, wird die Wahl des technischen Verfahrens dem Fachverbund überlassen; die Plattform stellt einen wiederverwendbaren PDP bereit, macht aber keine verbindlichen Vorgaben.

ADR-014	Ownership von grobgranularen Berechtigungen	Klärt, welche Entität Eigentümer grobgranularer Berechtigungen ist. Entschieden wurde, dass die Organisation die Ownership trägt, da nur Organisationen als stabile verantwortliche Akteure im Verwaltungsumfeld dauerhaft identifizierbar sind und Nutzer sowie Systeme stets als Vertreter einer Organisation auftreten.
ADR-015	System für Identifizierung von natürlichen Personen für das FöPD	Klärt, welches System zur Identifizierung natürlicher Personen im FöPD verwendet wird. Entschieden wurde für die BundID, da sie ein etabliertes staatliches Identifikationssystem ist, keine eigene Identity-Infrastruktur erfordert und europäische eID-Identitäten über eIDAS bereits angebunden sind.
ADR-016	System für Identifizierung von juristischen Personen für das FöPD	Klärt, welches System zur Identifizierung juristischer Personen im FöPD verwendet wird. Entschieden wurde für Mein Unternehmenskonto, da es ein verwaltungsnaher, breit anschlussfähiger Identitätsanker für juristische Personen ist und Governance sowie Datenschutz im staatlichen Verantwortungsbereich liegen.
ADR-017	Risikoadressierung von unzulässigen Veränderungen von grobgranularen Berechtigungen	Klärt, wie das Risiko unzulässiger Veränderungen grobgranularer Berechtigungen adressiert wird. Entschieden wurde für einen Transparency Log, der Berechtigungsänderungen manipulationssicher protokolliert und für föderierte Betreiber dezentral prüfbar macht, ohne zusätzliche zentrale Kontrollinstanzen zu schaffen.
ADR-018	Vorgaben für dezentrale Komponenten der Basisdienste	Klärt, für welche dezentralen Komponenten der Basisdienste verbindliche Architekturvorgaben der Plattform gelten. Entschieden wurde für einen gezielten Ansatz: Vorgaben betreffen ausschließlich die plattformkritischen und sicherheitsrelevanten Komponenten, um Fachverbünde nicht übermäßig zu belasten.
ADR-019	Scope der zentralen Nutzerautorisierung	Klärt, für welche Nutzergruppen die zentrale Nutzerautorisierung initial bereitgestellt wird. Entschieden wurde, den Scope zunächst auf Verwaltungskunden zu begrenzen, da bestehende staatliche Identitätslösungen unmittelbar angebunden werden können; eine spätere Erweiterung auf Behördenmitarbeitende bleibt möglich.
ADR-020	Logging von personenbezogenen Daten im Transparency Log	Klärt, ob und in welcher Form personenbezogene Daten im zentralen Transparency Log gespeichert werden. Entschieden wurde, das zentrale Log vollständig personen- und organisationsfrei zu halten; personenbezogene Nachvollziehbarkeit erfolgt ausschließlich in lokalen, geschützten Audit-Logs der jeweiligen Betreiberorganisation.
ADR-021	Transparency Log	Klärt, wie sicherheitsrelevante Zustände und Änderungen innerhalb der föderalen API-Autorisierungsinfrastruktur revisionssicher, manipulationsgeschützt und organisationsübergreifend überprüfbar

		dokumentiert werden. Entschieden wurde für den Einsatz eines kryptografisch verketteten Append-Only-Transparency-Logs auf Basis einer etablierten Transparenzplattform, da diese unabhängige Verifizierbarkeit, föderale Auditierbarkeit und ein flexibles, domänenspezifisch erweiterbares Datenmodell ermöglicht, ohne zusätzliche zentrale Kontrollinstanzen zu schaffen.
ADR-022	Security-Events – Format und Schnittstelle	Klärt, in welchem Format und über welche Schnittstelle sicherheitsrelevante Ereignisse innerhalb der föderalen API-Autorisierungsinfrastruktur organisationsübergreifend ausgetauscht werden. Entschieden wurde für das Shared Signals Framework (SSF) in Kombination mit signierten Security Event Tokens (SET), da damit ein etablierter, interoperabler und föderationsfähiger Standard zur Verfügung steht, der speziell auf Ereignisse im Identitäts- und Autorisierungskontext ausgerichtet ist und integritätsgesicherte Security-Events ermöglicht.
ADR-023	Verteilung von Policies und Attributen an verteilte Policy Decision Points (PDP)	Klärt, wie zentral verwaltete Policies und Attribute von Policy Retrieval Point (PRP) und Policy Information Point (PIP) an dezentrale Policy Decision Points (PDP) verteilt werden, sodass diese auch bei temporärer Nichtverfügbarkeit zentraler Dienste entscheidungsfähig bleiben. Entschieden wurde für ein selektives Polling-Modell mit lokalem Caching und optionalem Long-Polling, da es eine skalierbare, netzwerktopologieunabhängige Verteilung ermöglicht, Datenminimierung unterstützt und den PDP einen „last known good state“ ohne Inbound-Abhängigkeiten sicherstellt.

4.2 Übersicht der Systemlandschaft

Das vorliegende Kapitel gibt einen Überblick über die Systemlandschaft der föderalen API-Autorisierungsinfrastruktur. Die beteiligten Systeme werden in vier Gruppen unterteilt, die ihre jeweilige Rolle innerhalb der Gesamtarchitektur widerspiegeln: Infrastrukturnutzende Systeme sind die Nutzer und Betreiber der Infrastruktur; unterstützende externe Systeme stellen notwendige Dienste bereit, ohne selbst Teil der Kerninfrastruktur zu sein; die zentralen Systeme der Kerninfrastruktur bilden das plattformweit betriebene Herzstück; und die dezentralen Systeme der Kerninfrastruktur übernehmen zentrale Aufgaben in der Betriebsverantwortung der einzelnen Basisdienste.

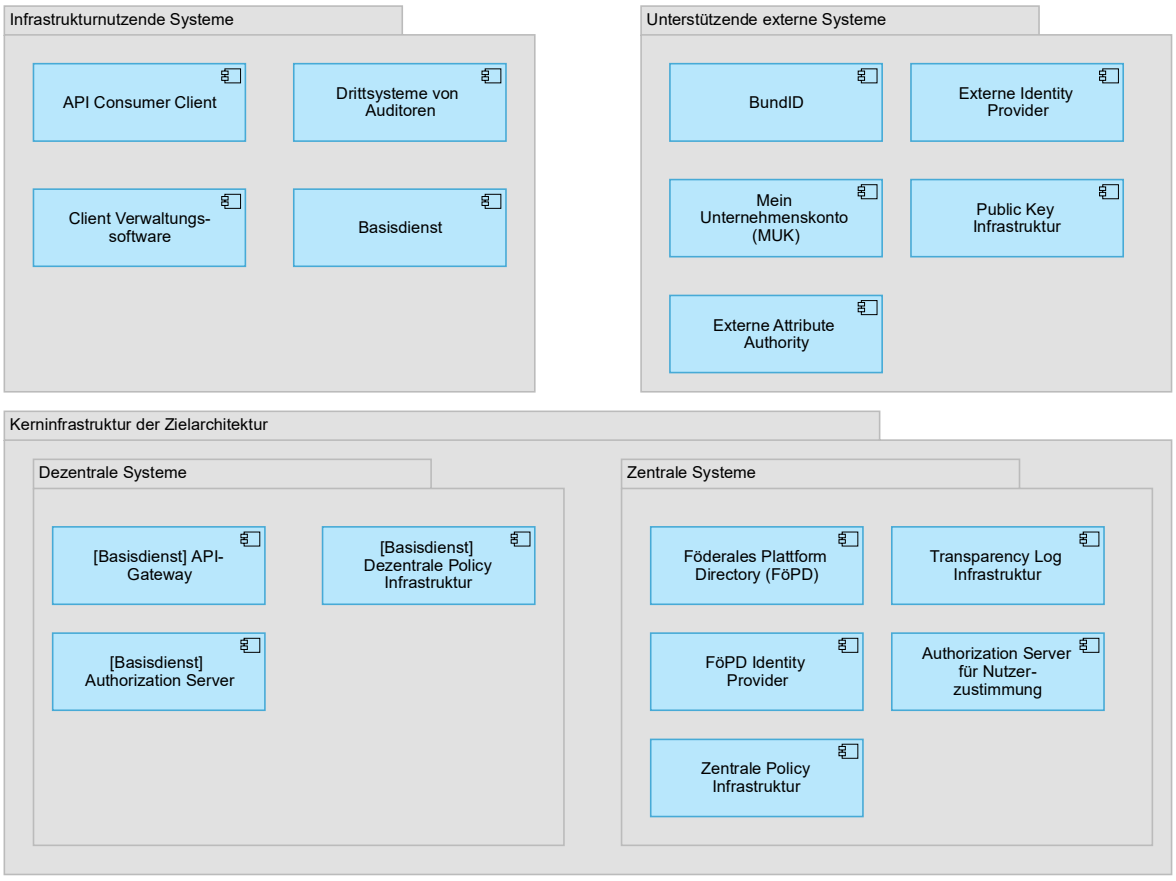


Abbildung 12: Übersicht der Systemlandschaft

4.2.1 Beschreibung der Systeme

4.2.1.1 Infrastrukturnutzende Systeme

Infrastrukturnutzende Systeme sind solche Systeme, die Services nutzen, die von der Kerninfrastruktur der Zielarchitektur bereitgestellt werden.

Tabelle 11: Infrastrukturnutzende Systeme

System	Kurzbeschreibung
API Consumer Client	Softwarekomponente einer fachlichen Anwendung, die als OAuth 2.0 Client auf Basisdienst-APIs zugreift.
Client Verwaltungssoftware	Software zur Verwaltung von API-Consumer-Anwendungen und deren Clients im FöPD. Sie ermöglicht die Anlage von Software-Einträgen für Basisdienst-Angebote, die der Organisation aktiv zur Nutzung zur Verfügung stehen, sowie den Empfang und die Verwaltung von Software Statement Assertions und Client-Schlüsseln.
Basisdienst	Fachliche Anwendung oder Infrastrukturkomponente, die APIs für nachnutzende Stellen bereitstellt und über die dezentrale Kerninfrastruktur (Authorization Server, API-Gateway, Policy-Infrastruktur) in die föderale API-Autorisierungsinfrastruktur eingebunden ist.
Drittsysteme von Auditoren	Externe Systeme, die Auditoren oder Prüfinstanzen den lesenden Zugriff auf den Transparency Log ermöglichen, um sicherheitskritische Ereignisse, Token-Ausstellungen und Berechtigungsentscheidungen der Infrastruktur nachzuvollziehen und zu prüfen.

4.2.1.2 Unterstützende externe Systeme

Unterstützende externe Systeme sind solche Systeme, die Services an Systeme der Kerninfrastruktur bereitstellen, die für die Umsetzung bestimmter System Use Cases notwendig sind. Diese Systeme werden aber selbst nicht umgesetzt und können nicht direkt verändert werden, jedoch können Anforderungen formuliert werden, wenn diese Systeme die benötigten Services nicht oder nicht in der gewünschten Ausprägung bereitstellen.

Tabelle 12: Unterstützende externe Systeme

System	Kurzbeschreibung
BundID	Zentrales Identitätsmanagementsystem des Bundes. Im Kontext der föderalen API-Autorisierungsinfrastruktur dient die BundID als externer Identity Provider zur Authentifizierung natürlicher Personen, insbesondere bei der Verifikation der handelnden Person im Organisationsregistrierungsprozess.
Mein Unternehmenskonto (MUK)	Plattform zur digitalen Identifikation juristischer Personen. Das MUK fungiert als externer Identity Provider zur Verifikation der Organisationsidentität im Registrierungsprozess und stellt SAML Assertions mit den Identitätsdaten der juristischen Person aus.
Externe Identity Provider	Externe Identitätsmanagementsysteme von Behörden oder anderen Organisationen, die dem FöPD Identity Provider als föderierte Quellen für

	Nutzeridentitäten dienen und Identitätsnachweise für natürliche Personen oder Organisationen ausstellen.
Public Key Infrastruktur	Externe PKI-Infrastruktur zur Ausstellung und Verwaltung von Zertifikaten, die für die kryptographische Absicherung der Plattformkommunikation benötigt werden. FöPD-Zertifikate bilden u. a. die Vertrauensbasis für die Signierung von Software Statement Assertions.
Externe Attribute Authority	Externe Stellen, die autoritative Attributnachweise für Organisationen oder natürliche Personen ausstellen (z. B. fachspezifische Behördenzertifizierungen oder Registrierungsnachweise). Diese Nachweise fließen als Teil des Nutzeridentitätsnachweises in das FöPD ein und können in Berechtigungsregeln referenziert werden.

4.2.1.3 Zentrale Systeme der Kerninfrastruktur

Zentrale Systeme der Kerninfrastruktur sind zentral betriebene Systeme, die für die Gesamtarchitektur Services bereitstellen.

Tabelle 13: Zentrale Systeme der Kerninfrastruktur

System	Kurzbeschreibung
Föderales Plattform Directory (FöPD)	Zentrales Self-Service-Portal und Single Source of Truth der föderalen API-Autorisierungsinfrastruktur. Das FöPD verwaltet die Registrierung von Organisationen, API-Consumer-Software, API-Clients und Plattformangeboten, stellt Software Statement Assertions aus, pflegt den Attributkatalog sowie die Berechtigungsregeln (Policies) und koordiniert die Bereitstellungs- und Freigabeprozesse für Plattformangebote.
FöPD Identity Provider	Zentraler Identity Provider der Plattform. Er verwaltet Nutzeraccounts und Organisationsattribute, authentifiziert FöPD-Nutzer und aggregiert Identitätsdaten aus externen Quellen (BundID, MUK, externe IdPs). Er dient als Vertrauensanker für die Nutzeridentitäten innerhalb der Plattform.
Zentrale Policy Infrastruktur	Zentrale Komponente für die Verwaltung und Bereitstellung von Berechtigungsregeln (Policies) und Policy-Entscheidungsinformationen. Sie ist die kanonische Quelle für alle aktiven Policies und stellt die Datengrundlage bereit, die von den dezentralen Policy-Infrastrukturen der Basisdienste repliziert wird.
Authorization Server für Nutzerzustimmung	Zentraler OAuth 2.0 Authorization Server für Szenarien mit expliziter Nutzerautorisierung (Authorization Code Flow). Er stellt API-unspezifische Opaque Access Tokens mit Nutzerautorisierung aus, die anschließend über das Token Exchange Protokoll gegen basisdienst-spezifische Access Tokens beim lokalen Authorization Server eingetauscht werden.

Transparency Log Infrastruktur	Zentrale Infrastruktur zur manipulationssicheren Protokollierung sicherheitskritischer Ereignisse der föderalen API-Autorisierungsinfrastruktur, wie Token-Ausstellungen und Berechtigungsentscheidungen. Der Transparency Log dient als revisionssicherer Prüfpfad für Betreiber und Auditoren.
SSF-Monitoring-Infrastruktur	Zentrale Infrastruktur zur Aggregation, Auswertung und Weiterleitung sicherheitsrelevanter Ereignissignale über alle Basisdienste hinweg. Die SSF-Monitoring-Infrastruktur implementiert das Shared Signals Framework (SSF) der OpenID Foundation als zentralen Event-Receiver und -Broker: Sie empfängt Security Event Tokens (SETs) von den dezentralen Kerninfrastrukturkomponenten der Basisdienste (Authorization Server, API-Gateway, dezentrale Policy-Infrastruktur) sowie von den zentralen Infrastrukturkomponenten (FöPD, FöPD Identity Provider, zentrale Policy-Infrastruktur), korreliert diese ereignisübergreifend und stellt aggregierte Risikosignale anderen Systemkomponenten bereit. Darüber hinaus fungiert die SSF-Monitoring-Infrastruktur als SSF-Transmitter gegenüber angeschlossenen SIEM-Systemen und ermöglicht so die Integration in übergeordnete Sicherheitsüberwachungsplattformen.

4.2.1.4 Dezentrale Systeme der Kerninfrastruktur

Dezentrale Systeme der Kerninfrastruktur sind solche Systeme, die zentrale Aufgaben übernehmen, aber in einer dezentraler Betriebsverantwortung bei den Basisdienst Betriebsorganisationen liegen. Die Bereitstellung von Lösungen zur Realisierung dieser lokalen Systeme und alternativer Lösungsoptionen wird in Kapitel 5.5 näher beleuchtet.

Tabelle 14: Dezentrale Systeme der Kerninfrastruktur

System	Kurzbeschreibung
[Basisdienst] Authorization Server	Dezentral bei jedem Basisdienst betriebener OAuth 2.0 Authorization Server und lokaler Vertrauensanker des Basisdienstes. Er nimmt Token-Anfragen von API Consumer Clients entgegen, holt Berechtigungsentscheidungen beim lokalen PDP ein und stellt Access Tokens für berechtigte Clients aus.
[Basisdienst] API-Gateway	Dezentral bei jedem Basisdienst betriebene Komponente, die als Policy Enforcement Point (PEP) allen eingehenden API-Anfragen vorgelagert ist. Das API-Gateway prüft bei jeder Anfrage die kryptographische Gültigkeit und Verwendungsberechtigung des Access Tokens und sichert so die laufende Kommunikation zwischen API Consumer Client und Basisdienst-API ab.
[Basisdienst] Dezentrale Policy Infrastruktur	Dezentral bei jedem Basisdienst betriebener Policy Decision Point (PDP) und ein lokaler Attribute Store. Die Dezentrale Policy Infrastruktur wertet Berechtigungsanfragen des Basisdienst Authorization Servers und des API-Gateways anhand der replizierten Policies und Attributinformationen aus der

	zentralen Policy Infrastruktur aus und gibt Berechtigungsentscheidungen zurück. Dezentrale Policy Infrastruktur agiert zudem als direkter SSF-Receiver: Sie empfängt Risikosignale von der zentralen SSF-Monitoring-Infrastruktur und kann daraufhin Berechtigungsentscheidungen unmittelbar anpassen, wie etwa durch Verweigerung des API-Zugriffs auf Basis eines erhöhten Risikowerts einer Software oder einer Client-Sperrung. Dieser direkte Signalkanal ergänzt den im Berechtigungskonzept beschriebenen PIP-Auslieferungsweg und ermöglicht eine Reaktion, die unabhängig vom regulären Policy-Replikationszyklus erfolgt.
[Basisdienst] SSF-Transmitter-Adapter	Dezentral bei jedem Basisdienst betriebene Adapterkomponente, die sicherheits- und betriebsrelevante Ereignisse der lokalen Kerninfrastruktur (Authorization Server, API-Gateway, dezentrale Policy-Infrastruktur) als Security Event Tokens (SETs) gemäß Shared Signals Framework aufbereitet und an die zentrale SSF-Monitoring-Infrastruktur übermittelt. Der Adapter übersetzt basisdienst-interne Ereignisse in das föderale SSF-Eventprofil und schließt damit die Basisdienst-Komponenten an die übergreifende Risiko- und Ereignisüberwachung der Plattform an. Die Übermittlung erfolgt wahlweise im Push- (RFC 9835) oder Poll-Modus (RFC 8936).

4.3 Technische Informationsarchitektur

Das vorliegende Kapitel beschreibt die technische Informationsarchitektur der föderalen API-Autorisierungsinfrastruktur. Es legt dar, welche zentralen Datenobjekte im System existieren, welche Systeme für diese verantwortlich sind und wie Informationen zwischen den Systemkomponenten fließen.

Kapitel 4.3.1 gibt einen Überblick über die zentralen Datenobjekte und ordnet ihnen die relevanten Informationskonzepte sowie die genutzten Austauschformate zu. Kapitel 4.3.2 dokumentiert in Form einer Informationsverantwortungsmatrix, welche Systeme in welcher Rolle mit den jeweiligen Datenobjekten interagieren, und klärt in einem gesonderten Unterkapitel die Fälle, in denen die Ownership eines Datenobjekts bewusst auf mehrere Systeme verteilt ist. Kapitel 4.3.3 schließt mit einer Darstellung der zentralen übergreifenden Informationsflüsse auf konzeptuellem Niveau ab.

4.3.1 Übersicht der zentralen Datenobjekte

Tabelle 15: Zentrale Datenobjekte

Datenobjekt	Kurzbeschreibung	Relevante Informationskonzepte	Genutzte Austauschformate
Organisation	Juristische Person, die als betriebsverantwortliche Stelle, Basisdienst nutzende Stelle, Betriebsorganisation Basisdienst oder Fachverbund verantwortliche Stelle im FöPD registriert ist.	Basisdienst Nutzer, Software Lieferant, Software Betreiber, Betriebsverantwortliche Stelle, Betriebsorganisation Basisdienst, Fachverbund verantwortliche Stelle, Freigebende Stelle, Plattformverantwortliche Stelle	SCIM (RFC 7643, RFC 7644)
Natürliche Person	Physische Person, die im Kontext einer Organisation im FöPD agiert, z. B. als Super-Admin oder delegierter Nutzer.	Basisdienst Nutzer	SCIM (RFC 7643, RFC 7644)
FöPD Account	Konto einer Organisation oder natürlichen Person im Föderalen Plattform Directory, das Zugang zu den Funktionen des FöPD gewährt und die zugewiesenen Nutzungsattribute enthält.	Basisdienst Nutzer, Software Lieferant, Software Betreiber, Betriebsverantwortliche Stelle, Betriebsorganisation Basisdienst, Fachverbund verantwortliche Stelle, Freigebende Stelle, Plattformverantwortliche Stelle	SCIM (RFC 7643, RFC 7644)
FöPD Zertifikat	Vertrauensanker für Software Statements, die vom FöPD herausgegeben werden.		X.509 (RFC 5280)
Policy	Regelwerk für Access Policies	Berechtigungskonzept	Offen: Potenzielle Spezifikationen sind DMN, OPA oder Cedar.
Policy Entscheidungsinformationen	Zusammenstellung relevanter Datensätze für die Prüfung von Bedingungen einer	API-Berechtigung, Anwendungsberechtigung	

	Policy wie bspw. Attribute einer Person, Organisation oder Software oder auch entscheidungsrelevante Kontextinformationen.		
Policy Entscheidung	Zugriffsentscheidung auf Basis einer Policy und Policy Entscheidungsinformationen.	API-Berechtigung, Anwendungsberechtigung	
Risikowert	Ein Risikowert in Bezug auf ein Subjekt in der Infrastruktur auf Basis von Sicherheitsereignissen, die mit dem Subjekt verbunden sind.		
Kataloginformationen von Plattformangeboten	Metadaten eines Plattformangebots eines Basisdienstes, die im FöPD veröffentlicht werden und die verfügbaren APIs, Berechtigungsmodelle und Nutzungsbedingungen beschreiben.	Plattformangebot, Basisdienst Angebot	
API-Consumer Software	Softwareprodukt einer betriebsverantwortlichen Stelle, das APIs von Basisdiensten nutzt und im FöPD registriert ist. Grundlage für die Ausstellung eines Software Statements.	API-Consumer	Software Statement (RFC 7591)
API-Client	Technische Instanz einer API-Consumer Software, die für den Abruf von Access Tokens beim OAuth Server registriert ist.	API-Consumer	Software Statement (RFC 7591)
Transparency Log	Fälschungssicheres, Merkle-Tree-basiertes Protokoll aller sicherheitsrelevanten Systemaktivitäten, das die Nachvollziehbarkeit und Auditierbarkeit der Plattform gewährleistet.		tlog-tiles (c2sp.org/tlog-tiles)

Client Authentifizierungsmittel	Kryptografisches Mittel, mit dem sich ein API-Client gegenüber dem OAuth Server authentifiziert.	Berechtigungskonzept	Private Key JWT (RFC 7523)
Access Token	Nachweis der erfolgreichen Authentifizierung eines API-Clients gegenüber dem OAuth Server, Grundlage für die Ausstellung eines Access Tokens.	Berechtigungskonzept	DPoP-bound Access Token – Opaque und JWT (RFC 9449, RFC 9068)
Nutzer Authentifizierungsnachweis	Nachweis der erfolgreichen Authentifizierung einer natürlichen Person gegenüber einem Identitätsprovider (BundID oder MUK).	Berechtigungskonzept	SAML Assertion (SAML 2.0) ID Token (OIDC Core 1.0)
Software Statement Assertion	Nachweis der Identität einer API-Consumer Software, ausgestellt auf Basis einer registrierten Software im FöPD und signiert durch das FöPD.	Berechtigungskonzept	Software Statement (RFC 7591)
Nutzer Identitätsnachweis	Nachweis der Identität einer natürlichen Person, der nach erfolgreicher Authentifizierung ausgestellt wird und Identitätsattribute enthält.	Berechtigungskonzept	ID Token (OIDC Core 1.0)
API-Ressource	Durch einen Basisdienst bereitgestellter API-Endpunkt, für dessen Nutzung eine gültige Berechtigung und ein entsprechender Access Token benötigt wird.	Ressource	

4.3.2 Informationsverantwortungsmatrix

4.3.2.1 Zweck und Abgrenzung

Die Informationsverantwortungsmatrix dient dazu, auf Enterprise-Architecture-Ebene kompakt und eindeutig zu dokumentieren, welche Systeme für welche Informationsobjekte verantwortlich sind und in welcher Form sie mit diesen interagieren. Sie ergänzt ArchiMate-Modelle um eine tabellarische Übersicht, die in Architektur-Reviews und Governance-Prozessen schnell konsumierbar ist.

Das klassische CRUD-Schema wird dabei bewusst nicht verwendet. Es vermischt die Dimension der Verantwortung (wer ist Eigentümer der Daten?) mit der Dimension der Interaktion (wer liest oder verändert die Daten, und auf welchem Weg?). Insbesondere die Unterscheidung zwischen einem System, das Daten direkt persistiert, und einem System, das eine Änderung über die Schnittstelle eines anderen Systems veranlasst, lässt sich in CRUD nicht ausdrücken.

Notation

Die Matrix verwendet vier Kürzel:

Tabelle 16: Informationsverantwortungsnotation

Kürzel	Bezeichnung	Bedeutung
O	Owner / System of Record	Das System ist autoritativer Eigentümer des Informationsobjekts. Es persistiert die Daten und stellt die maßgebliche Quelle dar. Pro Informationsobjekt darf es genau einen Owner geben.
I	Initiiert	Das System veranlasst Änderungen am Informationsobjekt über die Schnittstelle des Owners. Die Verantwortung für die Daten verbleibt beim Owner.
R	Read (direkt)	Das System liest das Informationsobjekt direkt beim Owner, typischerweise über eine synchrone Abfrage oder API-Aufruf.
S	Subscribe (repliziert)	Das System hält eine lokale Kopie des Informationsobjekts, die asynchron vom Owner bezogen wird (z.B. über Events, Feeds oder Replikationsmechanismen).
M	Manuell	Die Informationsübertragung erfolgt durch einen menschlichen Akteur. Es existiert kein technischer Integrationskanal zwischen den beteiligten Systemen.

Leseregeln

Mehrere O in einer Zeile signalisieren, dass die Ownership eines Informationsobjekts auf mehrere Systeme verteilt ist. Dies ist architektonisch zulässig, wenn die Verantwortlichkeit auf Attributebene eindeutig aufgeteilt werden kann – etwa, weil unterschiedliche Systeme autoritative Quellen für unterschiedliche Teilmengen der Attribute sind. Solche Fälle sind kein struktureller Fehler, sondern ein bewusstes Architekturmuster, das in Kapitel 4.3.2.3 für jedes betroffene Datenobjekt explizit aufgelöst wird. Fehlt dagegen jedes O in einer Zeile, ist die Ownership ungeklärt und muss adressiert werden.

I ohne eigenes O in derselben Zeile bedeutet, dass das System Änderungen delegiert. Es trägt keine Datenverantwortung, beeinflusst aber den Zustand des Informationsobjekts über die Schnittstelle des Owners.

R vs. S unterscheidet zwei technisch und architektonisch relevante Lesemuster: R impliziert eine direkte Abhängigkeit zur Verfügbarkeit des Owners; S signalisiert eine Entkopplung durch lokale Kopie, bringt aber Konsistenzfragen mit sich. Diese Unterscheidung sollte nur eingeführt werden, wenn sie für Verfügbarkeits- oder Konsistenzentscheidungen in der Architektur relevant ist.

M signalisiert, dass kein technischer Integrationskanal existiert und ein Mensch als Übertragungsglied fungiert. Dies ist architektonisch besonders relevant, da manuelle Übertragungen fehleranfällig, schwer auditierbar und nicht automatisierbar sind. Ein M in der Matrix ist damit ein unmittelbarer Hinweis auf ein potenzielles Integrations- oder Risikodefizit. Da ein manueller Schritt streng genommen einen Business Actor und keinen Systemkanal beschreibt, sollte das Kürzel M in der Legende entsprechend kenntlich gemacht werden.

R/S bzw. M/R kennzeichnen Einträge, bei denen der konkrete Zugriffstyp zum Zeitpunkt der Dokumentation noch nicht abschließend festgelegt wurde. R/S bedeutet, dass noch unklar ist, ob das System direkt beim Owner liest (R) oder eine lokal replizierte Kopie hält (S) – die Entscheidung hängt von noch offenen Verfügbarkeits- und Konsistenzanforderungen ab. M/R bedeutet, dass noch unklar ist, ob der Zugriff über einen technischen Kanal (R) oder manuell (M) erfolgt. Einträge dieser Art sind als offene Architekturpunkte zu verstehen, die im Zuge der Detailkonzeption zu klären und durch ein eindeutiges Kürzel zu ersetzen sind.

Leere Zellen bedeuten keine Beziehung zwischen dem System und dem Informationsobjekt.

Beispielmatrix

Tabelle 17: Beispiel einer Informationsverantwortungsmatrix

Informationsobjekt	CRM	ERP	Portal	Analytics	Lieferan-ten-Portal
Kundenstammdaten	O	I, R	I, R	S	
Bestelldaten	R	O	I, R	S	
Produktkatalog	R	R	O	R	
Rechnungsdaten	I, R	O	R	S	
Lieferanten-Preiskatalog		M			O

Am Beispiel der Kundenstammdaten ist ablesbar: Das CRM ist der autoritative Eigentümer. ERP und Portal können Änderungen veranlassen, lesen die Daten aber ebenfalls direkt. Analytics hält eine replizierte Kopie für Auswertungszwecke und ist damit vom CRM entkoppelt. Der Lieferanten-Preiskatalog verdeutlicht den Einsatz von M: Das Lieferanten-Portal ist Owner, das ERP erhält die Daten jedoch ausschließlich über manuelle Pflege durch einen Mitarbeiter. Es existiert kein technischer Integrationskanal, was die Aktualität der Preisdaten direkt vom manuellen Handeln abhängig macht und einen typischen Ausgangspunkt für eine spätere Automatisierung darstellt – etwa über eine Lieferanten-API oder einen EDI-Mechanismus.

4.3.2.2 Matrix

Tabelle 18: Informationsverantwortungsmatrix

Systeme O = Owner / System of Record I = Initiiert R = Read (direkt) S = Subscribe (repliziert) M = Manuell Datenobjekte	Föderales Plattform Directory	FöPD Identity Provider	Zentrale Policy Infrastruktur	Transparency Log	SSF-Monitoring-Infrastruktur	Authorization Server für Nutzerzustimmung	Basisdienst	[Basisdienst] Dezentrale Policy Infrastruktur	[Basisdienst] Authorization Server	[Basisdienst] API-Gateway	[Basisdienst] SSF-Transmitter-Adapter	BundID	Mein Unternehmenskonto (MUK)	Externe Identity Provider	Externe Attribute Authority	Public Key Infrastruktur	API Consumer Client	Externe Client Verwaltungssoftware	Drittsysteme von Auditoren
Organisation	I	O R S				R							O	O	O				
Natürliche Person		R				R						O		O	O				
FöPD Account	I	O																	
FöPD Zertifikat	I															O			
Policy			O					O S											
Policy Entscheidungsinformationen			O					O S											
Policy Entscheidung							R	O	R	R									
Risikowert			R/S		O			R/S											
Kataloginformationen von Plattformangeboten	O		R/S														R		
API-Consumer Software	O																R		

API-Client	O								O								R		
Transparency Log	I	I	I	O		I		S	I										S
Client Authentifizierungsmittel	R					R			R								O		
Access Token						O	R		O	R							R		
Nutzer Authentifizierungsnachweis	R	O				R	R			R		O	O	O			R		
Software Statement Assertion	O								R								R	R	
Nutzer Identitätsnachweis (Attestation)	R/S	O				R	R			R		O	O	O	O		R		
API-Ressource							O										I R		

4.3.2.3 Klärung geteilter Verantwortlichkeiten

Die föderale API-Autorisierungsinfrastruktur ist durch eine verteilte Systemlandschaft geprägt, in der bestimmte Datenobjekte keinem einzelnen System exklusiv zugeordnet werden können. Für diese Datenobjekte gilt, dass ihre Ownership entweder auf Attributebene aufzuteilen ist oder durch vertragliche sowie technische Mechanismen koordiniert werden muss. Das vorliegende Kapitel identifiziert die betroffenen Datenobjekte und beschreibt die architektonisch gewählte Strategie zur Klärung der Verantwortlichkeit.

Eine geteilte Ownership entsteht in der Infrastruktur aus zwei wesentlichen Mustern: Erstens aus der föderalen Systemstruktur, bei der gleichartige Komponenten dezentral für jeden Basisdienst existieren (z. B. Authorization Server oder Policy-Infrastrukturen), aber funktional denselben Typ von Datenobjekt erzeugen. Zweitens aus der Nutzung externer Identitätsprovider (BundID, MUK, externe IdPs), die jeweils autoritative Quellen für bestimmte Teilmengen von Identitätsdaten darstellen. Die nachfolgende Tabelle fasst die betroffenen Datenobjekte und die jeweils gewählte Strategie zur Klärung der Verantwortlichkeit zusammen.

Tabelle 19: Strategie zur Datenverantwortlichkeit

Datenobjekt	Strategie zur Datenverantwortlichkeit
Organisation	Attributbasierte Aufteilung: Der FöPD Identity Provider ist Owner der für die Plattformnutzung relevanten Organisations-Attribute (FöPD-spezifische Attribute, Rollen, Account-Status) und fungiert als System of Record der Plattform. BundID, MUK und externe Identity Provider sind Owner der Identitätsnachweise der juristischen Person in ihrem jeweiligen Geltungsbereich. Der FöPD IDP aggregiert und normalisiert diese eingehenden Nachweise.
Natürliche Person	Attributbasierte Aufteilung: BundID ist Owner der Identitätsnachweise für natürliche Personen im Kontext der öffentlichen Verwaltung (eID-basiert). Externe Identity Provider sind Owner für ihren jeweiligen Geltungsbereich, wie bspw S.A.F.E für die Justiz oder ausländische eID-Provider für Staatsbürger im EU-Ausland. Der FöPD Identity Provider ist System of Record für plattformspezifische Nutzerattribute (Rollen und Berechtigungen innerhalb des FöPD).
Policy / Policy Entscheidungsinformationen	Koordination über Hierarchie: Die Zentrale Policy Infrastruktur ist Owner der kanonischen Policy-Definition sowie der Policy Entscheidungsinformationen und damit autoritativer Ursprung aller Regeln. Die dezentrale Policy Infrastruktur des Basisdienstes hält eine lokal replizierte Kopie (Subscribe), die für Entscheidungsfähigkeit im laufenden Betrieb ohne Zentralabhängigkeit erforderlich ist.

	Die dezentrale Policy Infrastruktur kann zusätzliche Policies definieren, die auf feingranularer Ebene greifen, aber diese dürfen nicht im Konflikt mit den übergeordneten Regeln auf grobgranularer Ebene stehen.
API-Client	Attributbasierte Aufteilung: Das FöPD ist Owner der plattformweiten Registrierung der API-Consumer Software und der assoziierten Software Statement Assertions. Der [Basisdienst] Authorization Server ist Owner der lokal relevanten Client-Registrierung (OAuth 2.0 Dynamic Client Registration) und vergibt bspw. die lokal gültige Client-ID. Beide Repräsentationen desselben Clients sind über die Software Statement Assertion technisch verknüpft. Das FöPD gilt als führende Quelle.
Access Token	Geltungsbereichsbasierte Aufteilung: Der Authorization Server für Nutzerzustimmung ist Owner des plattformweiten Access Tokens mit Nutzerautorisierung (ohne API-Berechtigung). Der [Basisdienst] Authorization Server ist Owner des basisdienst-spezifischen Access Tokens, das durch Token Exchange oder direkte Ausstellung entsteht und ausschließlich für seinen Geltungsbereich gültig ist. Beide Token-Typen erfüllen unterschiedliche Zwecke und sind nicht substituierbar.
Nutzer Authentifizierungsnachweis	Herkunftsbasierte Aufteilung: Jeder Identity Provider (BundID, MUK, externe IdPs) ist Owner des Authentifizierungsnachweises, den er selbst ausstellt. Der FöPD Identity Provider validiert eingehende Assertions und ist System of Record für den plattforminternen Authentifizierungsnachweis. Nachrichten von Dritten bleiben in ihrem Ursprungsgeltungsbereich authoritativ; der FöPD IDP übersetzt sie in ein plattforminternes Format.
Nutzer Identitätsnachweis (Attestation)	Herkunftsbasierte Aufteilung mit Aggregation: Analog zum Authentifizierungsnachweis ist jeder ausstellende IdP Owner seines eigenen Identitätsnachweises. Zusätzlich ist die Externe Attribute Authority Owner von fachspezifischen Attributen in ihrem Geltungsbereich. Der FöPD Identity Provider aggregiert und normiert die Attribute aus verschiedenen Quellen und ist damit System of Record für den plattformrelevanten Nutzer Identitätsnachweis.

4.3.3 Zentrale übergreifende Informationsflüsse

Dieses Kapitel gibt einen konzeptuellen Überblick über die zentralen Informationsflüsse der föderalen API-Autorisierungsinfrastruktur. Die Darstellungen fokussieren auf die Informationsobjekte, die zwischen den Systemkomponenten ausgetauscht werden, und abstrahieren bewusst von prozessualen Details wie der konkreten Abfolge von Nutzerinteraktionen oder dem internen Verhalten einzelner Komponenten. Die detaillierte Beschreibung der IT-Unterstützung der Kernprozesse folgt in Kapitel 4.4.

4.3.3.1 High Level Informationsflüsse in der Gesamtinfrastruktur

Die Abbildung zeigt eine konzeptuelle Übersicht der zentralen Informationsflüsse zwischen den Systemkomponenten der föderalen API-Autorisierungsinfrastruktur. Sie verdeutlicht, welche Informationsobjekte in welcher Richtung zwischen den Hauptkomponenten ausgetauscht werden: der zentralen Infrastruktur (FöPD, FöPD Identity Provider, Zentrale Policy Infrastruktur, Transparency Log Infrastruktur, SSF Monitoring Infrastruktur, Authorization Server für Nutzerzustimmung), den föderalen Identity Providern (BundID, MUK, externe IdPs), der Externen Attribute Authority, der Public Key Infrastruktur, der dezentralen Betriebsumgebung des Basisdienstes sowie dem API Consumer Client. Die Darstellung dient als Orientierungsrahmen für die detaillierten Informationsflüsse in den nachfolgenden Unterkapiteln.

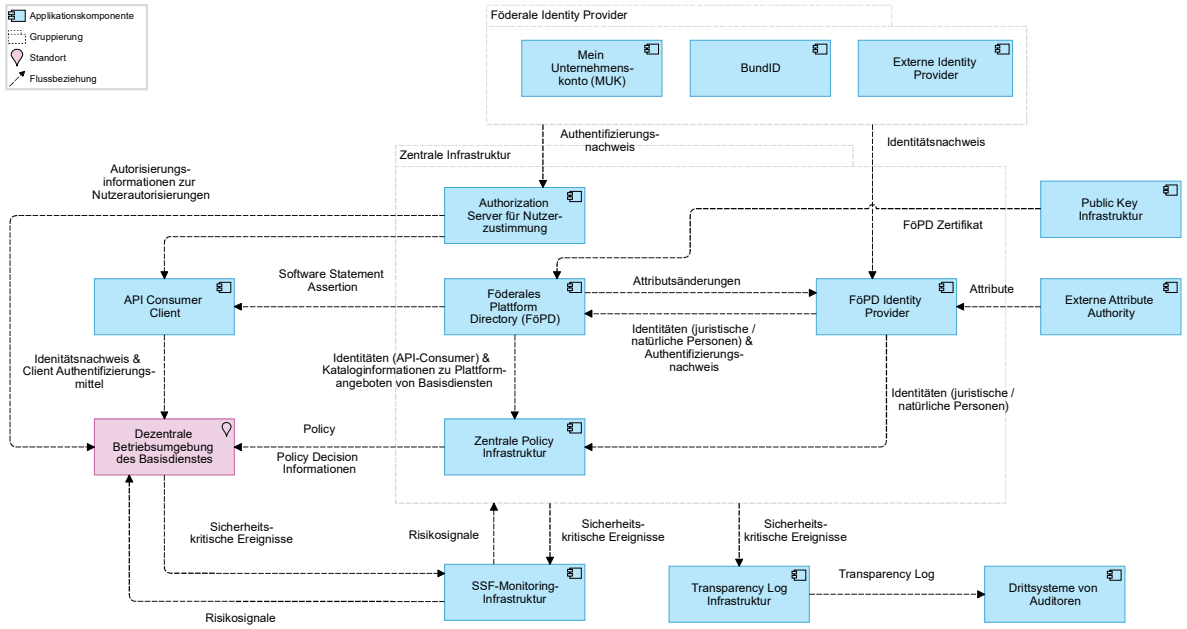


Abbildung 13: High Level Informationsflüsse in der Gesamtinfrastruktur

4.3.3.2 Detailinformationsflüsse im Kontext der API-Registrierung

Die Abbildung zeigt die Informationsflüsse im Kontext der API-Registrierung, d. h. der Vorbereitung eines API Consumer Clients für die technische Nutzung von Basisdienst-APIs. Zentrales Informationsobjekt ist die Software Statement Assertion (SSA): Das Föderale Plattform Directory stellt diese auf Basis der registrierten API-Consumer Software aus und stellt sie der Client Verwaltungssoftware sowie dem API Consumer Client zur Verfügung. Auf Basis der SSA registriert sich der API Consumer Client bei den relevanten Basisdienst Authorization Servern, die

damit die für den späteren Token-Austausch notwendigen Client-Metadaten erhalten. Ergänzend zeigt der Informationsfluss die Einbindung der Public Key Infrastruktur für die Bereitstellung von FöPD-Zertifikaten, die für die sichere Kommunikation mit der Plattform benötigt werden.

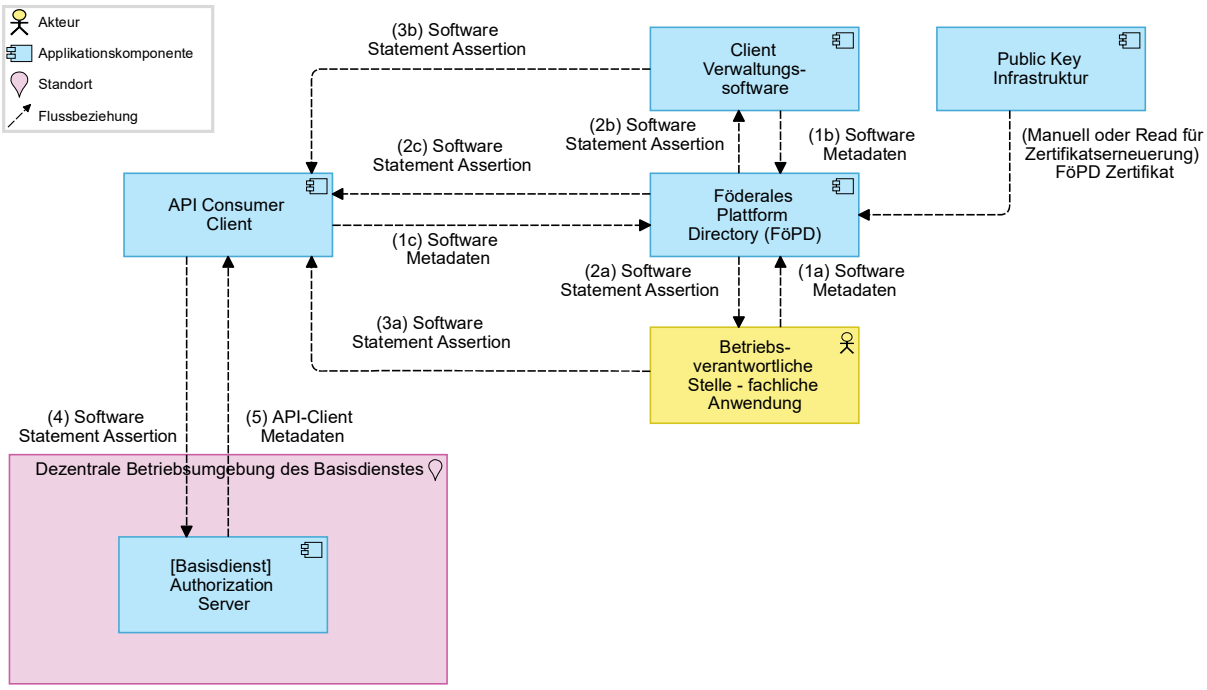


Abbildung 14: Detaildatenflussdiagramm für API-Registrierung

4.3.3.3 Detailinformationsflüsse im Kontext des API-Aufrufs

Die Abbildung zeigt die Informationsflüsse im Kontext eines API-Aufrufs, d. h. der Laufzeitautorisierung beim Zugriff eines API Consumer Clients auf eine API-Ressource eines Basisdienstes. Der API Consumer Client übermittelt seinen Opaque Access Token an das API-Gateway des Basisdienstes, das diesen gegen einen intern signierten Access Token eintauscht und eine Zugriffsevaluierung bei der dezentralen Policy Infrastruktur anfordert. Die dezentrale Policy Infrastruktur bezieht hierfür lokal replizierte Policies, Policy Entscheidungsinformationen sowie den Transparency Log von der zentralen Infrastruktur. Die resultierende Policy Entscheidung fließt sowohl in das API-Gateway als auch in den Basisdienst selbst ein und steuert damit den Zugriff auf die API-Ressource.

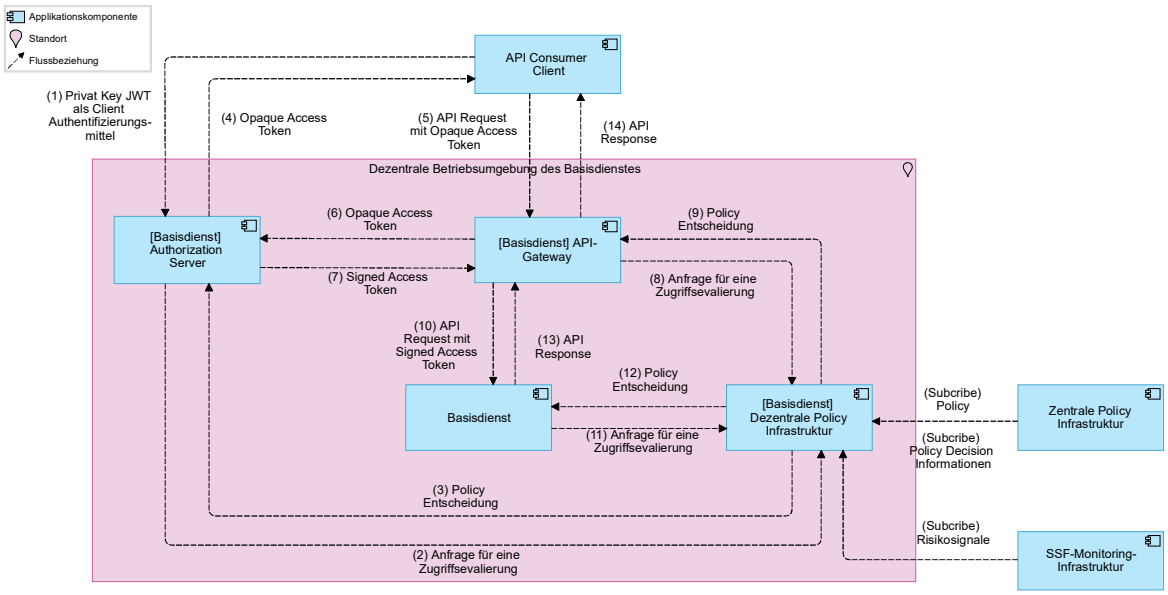


Abbildung 15: Detaildatenflussdiagramm für API-Aufruf

4.3.3.4 Detailinformationsflüsse im Kontext des Umtauschs von Nutzerautorisierungen in eine lokale API-Berechtigung

Die Abbildung zeigt die Informationsflüsse im Kontext des Token-Austauschs für Szenarien mit expliziter Nutzerautorisierung. In diesem Spezialfall holt sich ein API Consumer Client zunächst einen API-unspezifischen Opaque Access Token mit einer Nutzerautorisierung beim zentralen Authorization Server für Nutzerzustimmung. Da dieser Token keine Berechtigung für einen konkreten Basisdienst enthält, wird er über das Token Exchange Protokoll (RFC 8693) gegen einen basisdienst-spezifischen Opaque Access Token eingetauscht, der eine Nutzerautorisierung mit API-Berechtigung beinhaltet. Ab diesem Punkt ist der weitere Ablauf – einschließlich der Token-Validierung durch das [Basisdienst] API-Gateway, der Policy-Entscheidung und des finalen API-Aufrufs – identisch mit dem in Kapitel 4.3.3.3 beschriebenen Informationsfluss für den API-Aufruf.

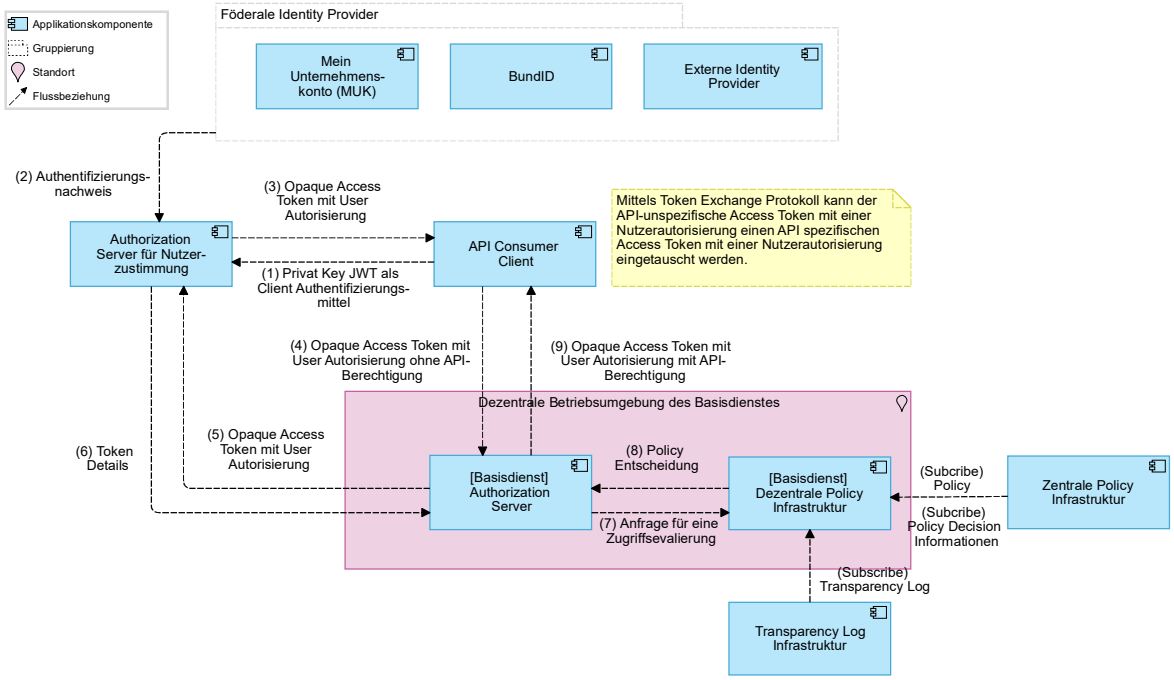


Abbildung 16: Detaildatenflussdiagramm für Token Austausch für lokalen API-Zugriff

4.4 Prozessübersicht und Darstellung der IT-Prozessunterstützung

Das vorliegende Kapitel beschreibt die Prozesslandschaft der föderalen API-Autorisierungsinfrastruktur und die IT-Unterstützung der identifizierten Kernprozesse. Kapitel 4.4.1 gibt einen Überblick über alle relevanten Kern- und Supportprozesse der Zielarchitektur und stellt deren Ablauflogik auf hohem Abstraktionsniveau dar. Kapitel 4.4.2 beschreibt die IT-Unterstützung jedes einzelnen Kernprozesses im Detail, einschließlich der beteiligten Use Cases, der relevanten Informationsflüsse und der Prozessabläufe. Ergänzende Hinweise zu den Supportprozessen, die im Rahmen dieses Konzepts nicht vollständig ausgearbeitet werden, schließen das Kapitel ab.

4.4.1 Prozessübersicht

Die nachfolgende Abbildung zeigt die Prozesslandkarte der föderalen API-Autorisierungsinfrastruktur. Sie unterscheidet zwischen Kernprozessen, die unmittelbar zur Erbringung des Plattformnutzens beitragen, und Supportprozessen, die den laufenden Betrieb der Infrastruktur unterstützen. Im Rahmen dieses Konzepts werden nur solche Prozesse benannt, die für das Verständnis und die Konzeption der Zielarchitektur relevant sind. Eine vollständige Betrachtung aller Betriebsprozesse ist nicht Gegenstand dieses Dokuments.

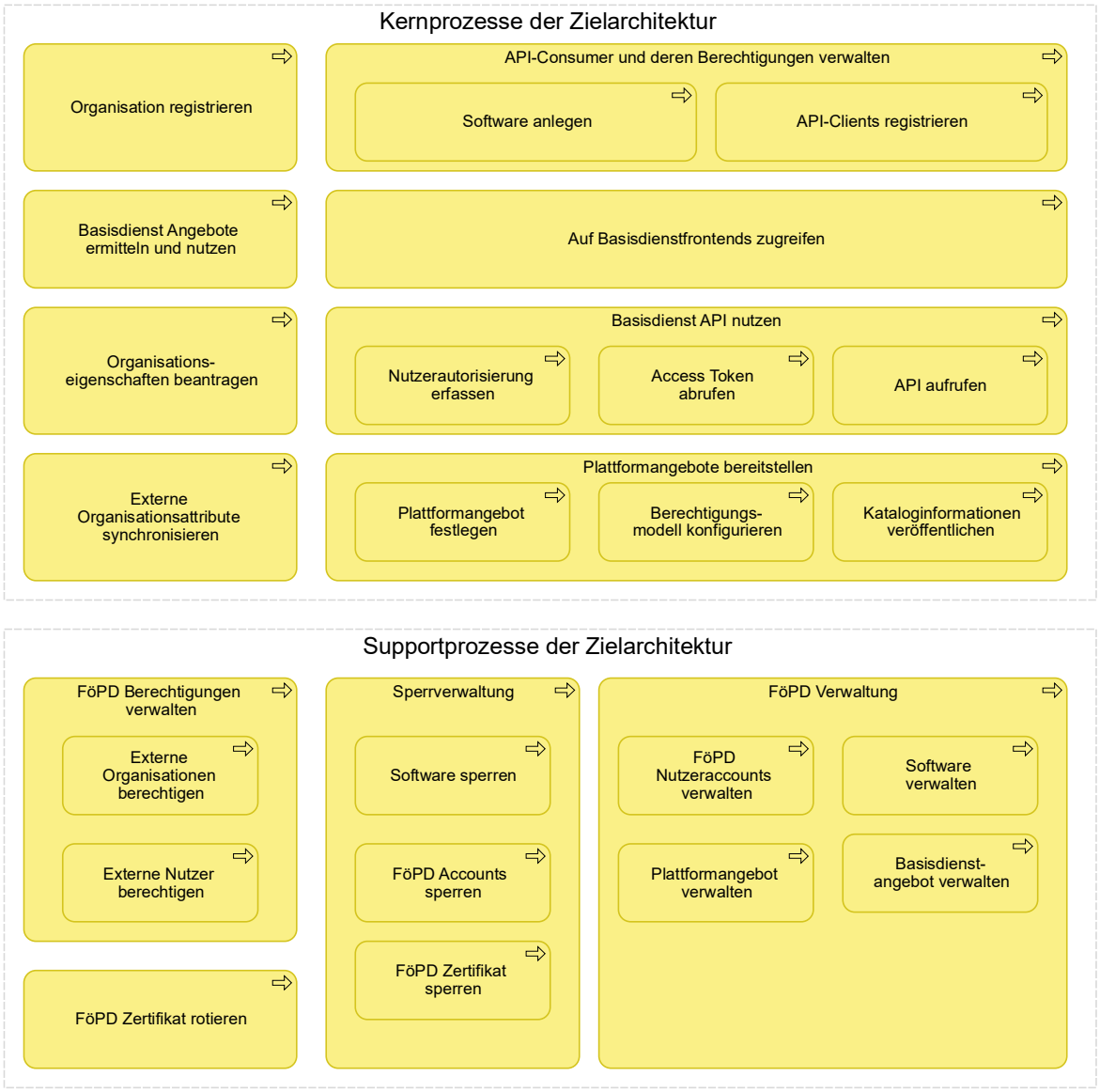


Abbildung 17: Landkarte der unterstützten Prozesse

Die nachfolgende Abbildung zeigt den High Level Prozessablauf der Kernprozesse und deren Abhängigkeiten untereinander. Ausgangspunkt aller Pfade ist der Prozess „Organisation registrieren", der eine Vorbedingung für alle weiteren Aktivitäten darstellt. Über den Prozess „Organisationseigenschaften beantragen" hinterlegt die Organisation die für die spätere Nutzung benötigten Eigenschaften. Auf dieser Grundlage verzweigen sich die Pfade. Auf der linken Seite steht der Pfad zur Nutzung von Basisdienst-Angeboten: Über den Prozess „Basisdienst Angebote ermitteln und nutzen" überführt die Organisation ein Angebot in die aktive Nutzung. Daran schließt sich entweder der Pfad zur API-Nutzung an, der über die Anlage einer Software

und die Registrierung von API-Clients bis hin zum API-Aufruf führt und optional um eine vorgelagerte Nutzerautorisierung ergänzt wird, oder der Pfad zum direkten Zugriff auf ein Frontend-Angebot. Auf der rechten Seite steht der Pfad zur Bereitstellung von Plattformangeboten, der von der Festlegung des Plattformangebots über die Konfiguration des Berechtigungsmodells bis zur Veröffentlichung der Kataloginformationen reicht.

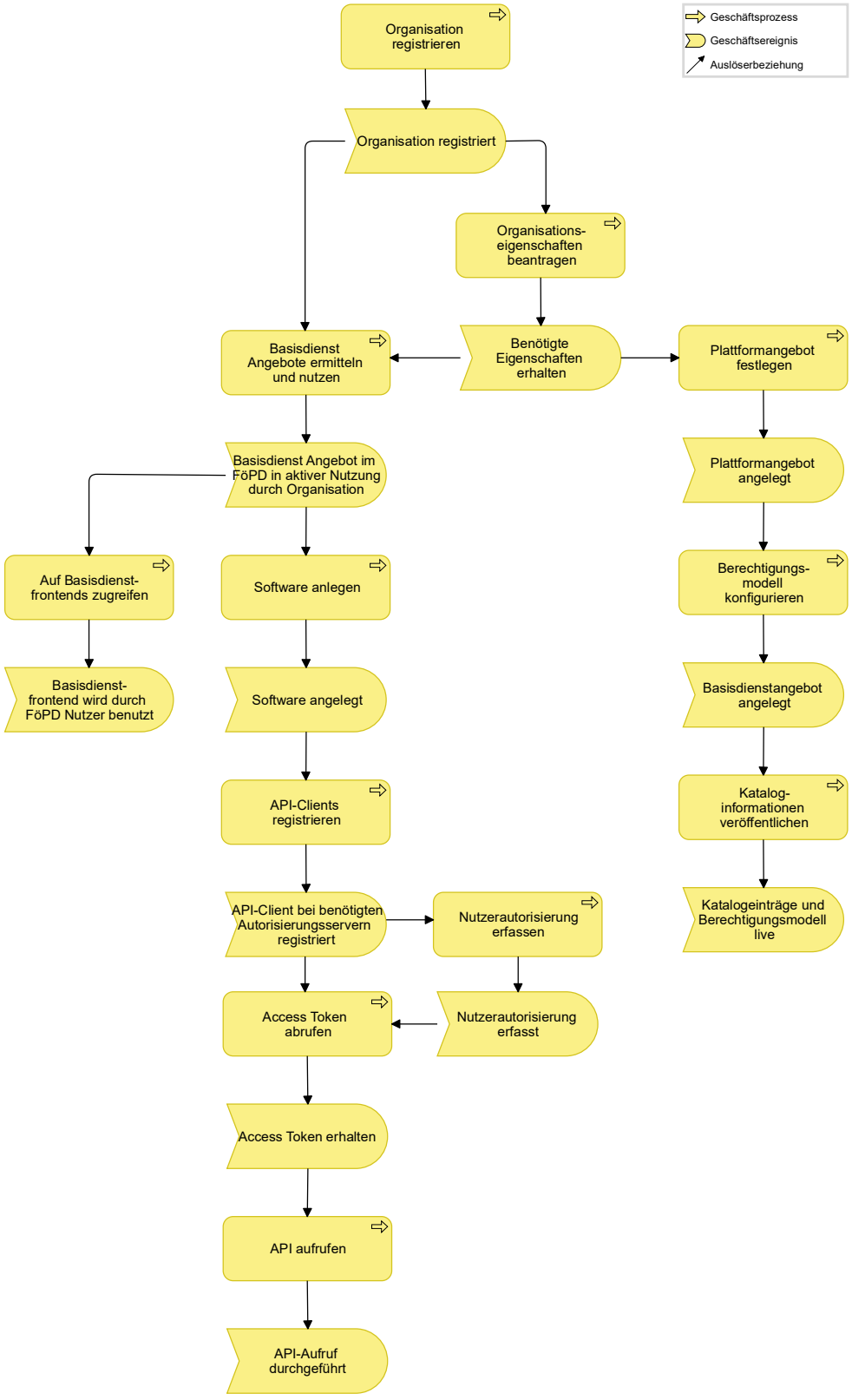


Abbildung 18: High Level Prozessablauf der Kernprozesse

4.4.2 Beschreibung der IT-Unterstützung der Kernprozesse

Dieses Kapitel beschreibt die IT-Unterstützung der Kernprozesse der föderalen API-Autorisierungsinfrastruktur im Detail. Für jeden Kernprozess werden die relevanten Use Cases, der Informationsfluss im Prozessablauf sowie die Ablaufbeschreibung mit Sequenzdiagrammen dargestellt. Die identifizierten Supportprozesse werden in diesem Konzept nicht im gleichen Detailgrad behandelt; ergänzende Hinweise zu ihnen finden sich in Kapitel 4.3.3. Die Use Cases der Kernprozesse werden darüber hinaus in Kapitel 4.4 in tabellarischer Form zusammengefasst und den umsetzenden Systemen sowie den genutzten Standardprotokollen zugeordnet.

4.4.2.1 Prozess „Organisation registrieren“

Der Prozess „Organisation registrieren" ist keiner übergeordneten Prozessgruppe zugeordnet und bildet die Grundvoraussetzung für alle drei Wertströme – „APIs von Basisdiensten integrieren", „APIs von Basisdiensten bereitstellen" sowie „Angebote von Basisdiensten nutzen" –, da eine Organisation erst nach erfolgreicher Registrierung im FöPD zur Wahrnehmung plattformgestützter Rollen berechtigt ist. Der Prozess wird durch das Föderale Plattform Directory als zentrales System umgesetzt, das für die Registrierung auf externe Identitätsprovider (MUK, BundID) sowie auf das eigene Identity Management zurückgreift. Er umfasst die Verifizierung der Identität der Organisation und der handelnden natürlichen Person, die Accountanlage sowie die initiale Konfiguration der Nutzungsrollen der Organisation im FöPD.

Relevante Use Cases:

Tabelle 20: Use Case der Kernprozesse

Use Cases	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Organisation registrieren	Föderales Plattform Directory (FöPD)	Organisation	Zentraler Use Case des Prozesses: Entgegennahme der Registrierungsanfrage, Koordination der Identitätsprüfung, Accountanlage und Rückmeldung des Ergebnisses an die Organisation.
FöPD Nutzer authentifizieren	FöPD Identity Provider	Föderales Plattform Directory (FöPD)	Authentifizierung eines beim FöPD registrierten Nutzers gegenüber dem FöPD-Frontend nach abgeschlossener Registrierung.

Identitäten überprüfen	FöPD Identity Provider	Föderales Plattform Directory (FöPD)	Überprüfung und Zusammenführung der Identitätsnachweise von Organisation und handelnder Person auf Basis der von BundID und MUK ausgestellten Assertions.
Accounts verwalten	FöPD Identity Provider	FöPD Identity Provider	Anlage des Organisationsaccounts und des initialen Super-Admin-Accounts sowie Speicherung und Aktualisierung der zugehörigen Attribute im Identity Provider.
Natürliche Personen authentifizieren	BundID	FöPD Identity Provider	Authentifizierung der handelnden natürlichen Person via BundID und Ausstellung einer SAML-Assertion mit den persönlichen Identitätsdaten.
Juristische Personen authentifizieren	Mein Unternehmenskonto (MUK)	FöPD Identity Provider	Authentifizierung der juristischen Person via MUK und Ausstellung einer SAML-Assertion mit den Identitätsdaten der Organisation.

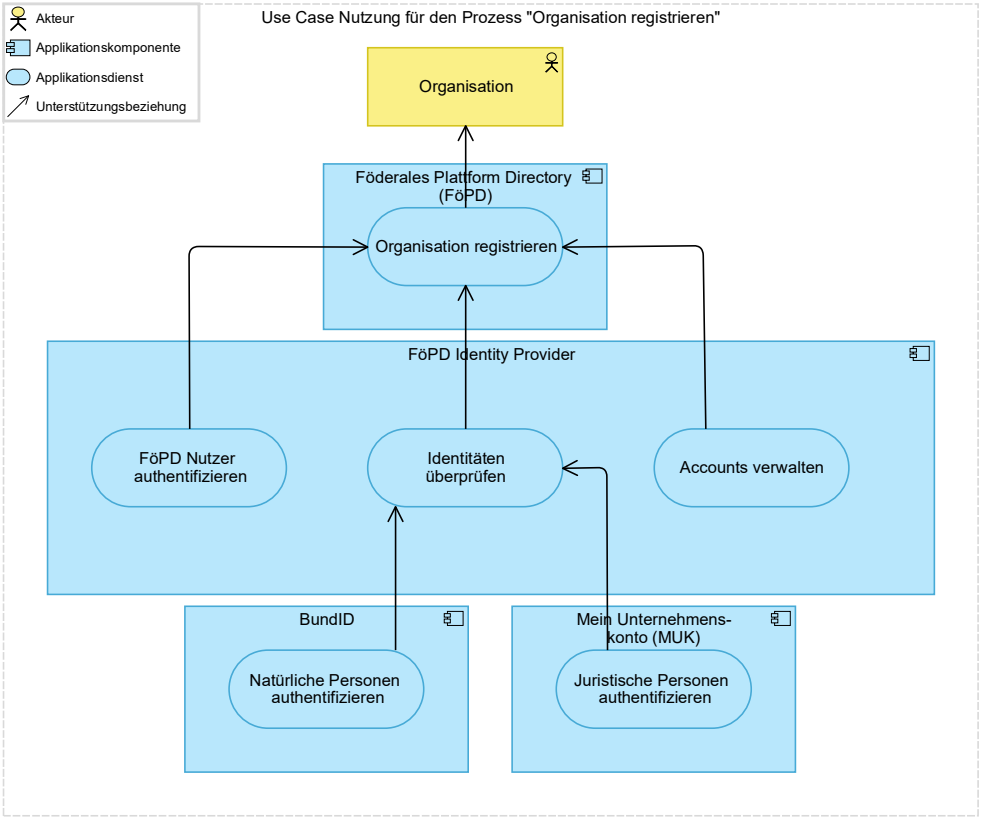


Abbildung 19: Use Nutzungsdiagramm für den Prozess „Organisation registrieren“

Informationsfluss im Prozessablauf:

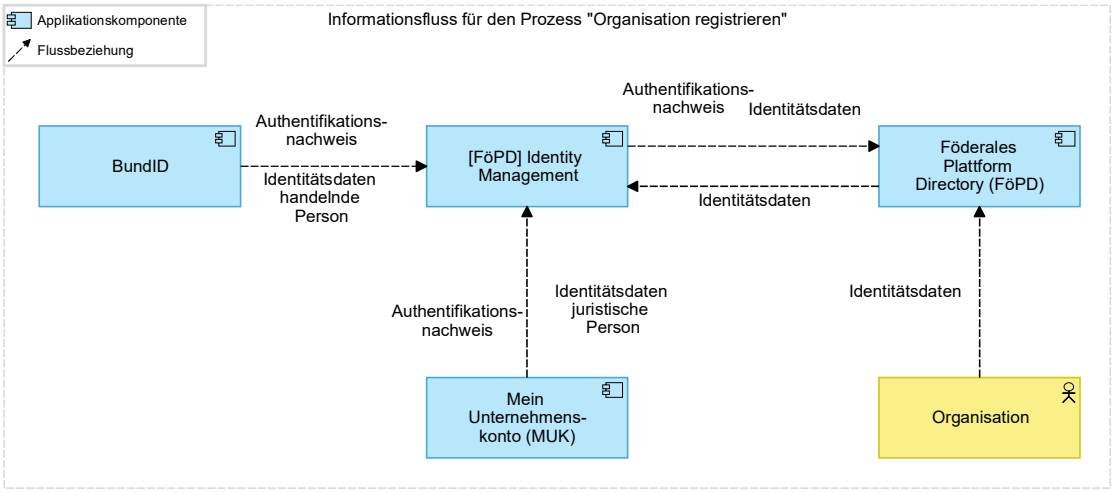


Abbildung 20: Informationsflussdiagramm für den Prozess „Organisation registrieren“

Ablaufbeschreibung

Teil 1: Registrierung

Der Prozess wird durch einen Nutzer der Organisation ausgelöst, der die Registrierung über das Self-Service-Portal des FöPD initiiert. Das FöPD leitet den Nutzer zunächst an den FöPD IDP weiter, der seinerseits eine Weiterleitung an das Mein Unternehmenskonto (MUK) initiiert, um die Identität der Organisation zu verifizieren. Der Nutzer meldet sich beim MUK an, das den Loginvorgang intern prüft und anschließend eine SAML-Assertion mit den Identitätsdaten der juristischen Person ausstellt. Diese Assertion wird über den Browser des Nutzers an das [FöPD] Identity Management weitergeleitet, das die Assertion prüft und auswertet, einen bestehenden Account für die Organisation überprüft und die relevanten Organisationsdaten übernimmt.

Im zweiten Authentifizierungsschritt leitet das [FöPD] Identity Management den Nutzer an die BundID weiter, um die Identität der handelnden natürlichen Person zu verifizieren. Der Nutzer meldet sich bei der BundID an, die ebenfalls eine SAML-Assertion ausstellt und den Nutzer zurück an das FöPD leitet. Das [FöPD] Identity Management prüft auch diese Assertion, übernimmt die Identitätsdaten der handelnden Person und übermittelt das gesamte Prüfungsergebnis samt Identitätsdaten an das FöPD.

Nachdem beide Identitäten verifiziert wurden, hat der Nutzer die Möglichkeit, die übermittelten Daten durch eigene Angaben zu ergänzen. Das FöPD zeigt dem Nutzer das Prüfungsergebnis sowie die Nutzungsbedingungen an. Nach Bestätigung der Nutzungsbedingungen übermittelt das FöPD die erfassten Daten an das [FöPD] Identity Management und initiiert die Accountanlage. Dort werden der Organisationsaccount sowie ein initialer Super-Admin-Account angelegt. Das FöPD veranlasst den Versand eines Briefes an die hinterlegte Adresse der Organisation, der über die erfolgte Accountanlage informiert und eine missbräuchliche Registrierung durch einzelne Mitarbeiter erkennbar macht. Abschließend zeigt das FöPD dem Nutzer das Ergebnis der Registrierung an.

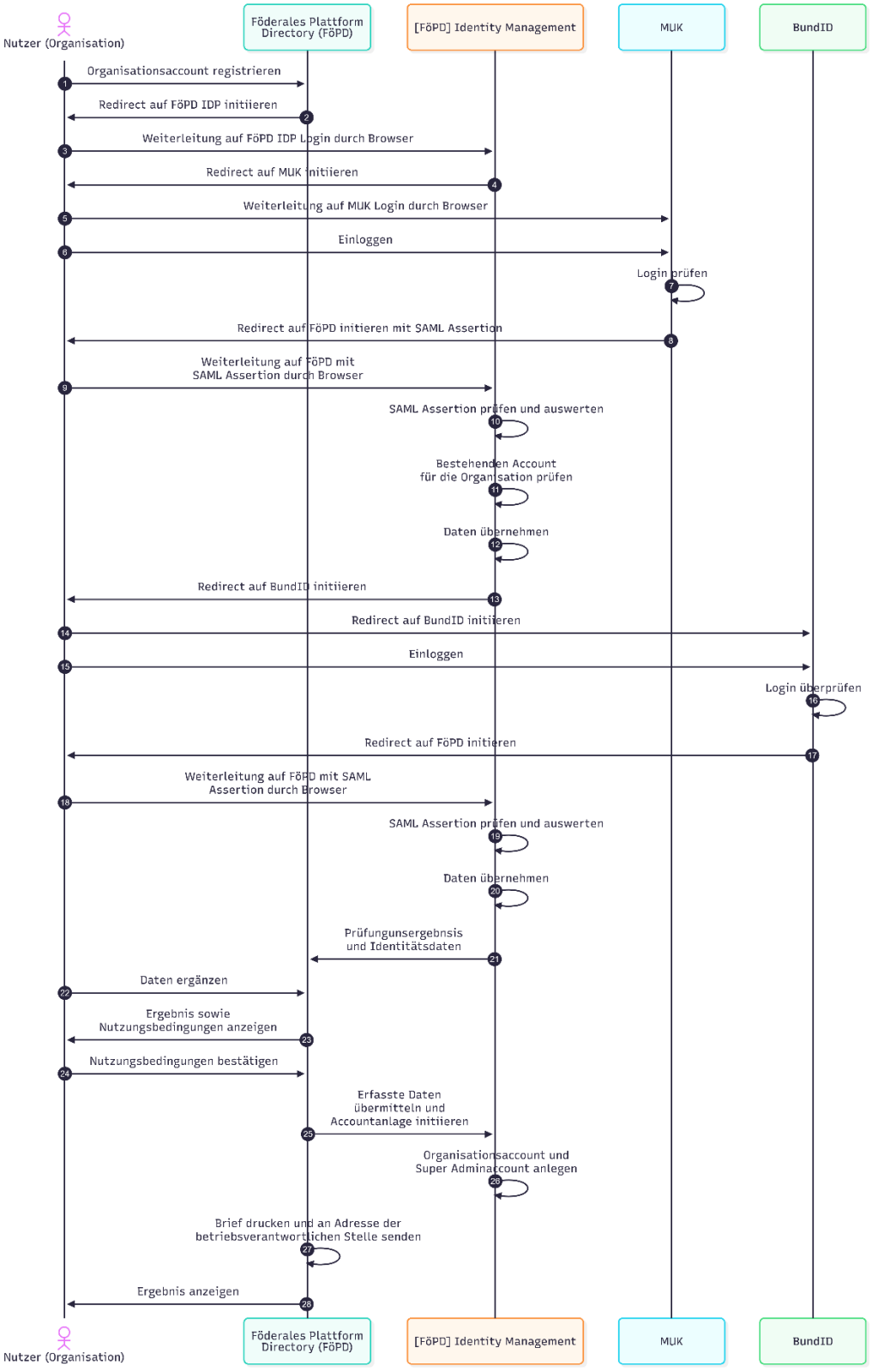


Abbildung 21: Sequenzdiagramm für den Prozess Organisation registrieren“ - Teil 1

Teil 2: Initialer Login und Nutzungskonfiguration

Nach abgeschlossener Registrierung meldet sich der Nutzer erstmalig am FöPD an. Das FöPD leitet den Nutzer an den FöPD IDP weiter, der den Login intern überprüft und anschließend einen Redirect auf das FöPD Frontend initiiert. Der Browser des Nutzers wird auf das FöPD Frontend weitergeleitet. Das FöPD prüft den erhaltenen ID-Token und gewährt dem Nutzer nach erfolgreicher Prüfung den Zugriff.

Im Anschluss zeigt das FöPD dem Nutzer die verfügbaren Nutzungsoptionen an. Die Organisation kann dabei zwischen verschiedenen Rollen wählen: Betriebsverantwortliche Stelle, Basisdienst nutzende Stelle, Betriebsorganisation Basisdienst sowie Fachverbund verantwortliche Stelle. Die gewählten Optionen schalten entsprechende Funktionen im FöPD frei; die Bereitstellung und Steuerung von Angeboten erfordern darüber hinaus gesonderte Freigaben. Nach der Auswahl hinterlegt das FöPD die entsprechenden Attribute zur Nutzung im [FöPD] Identity Management.

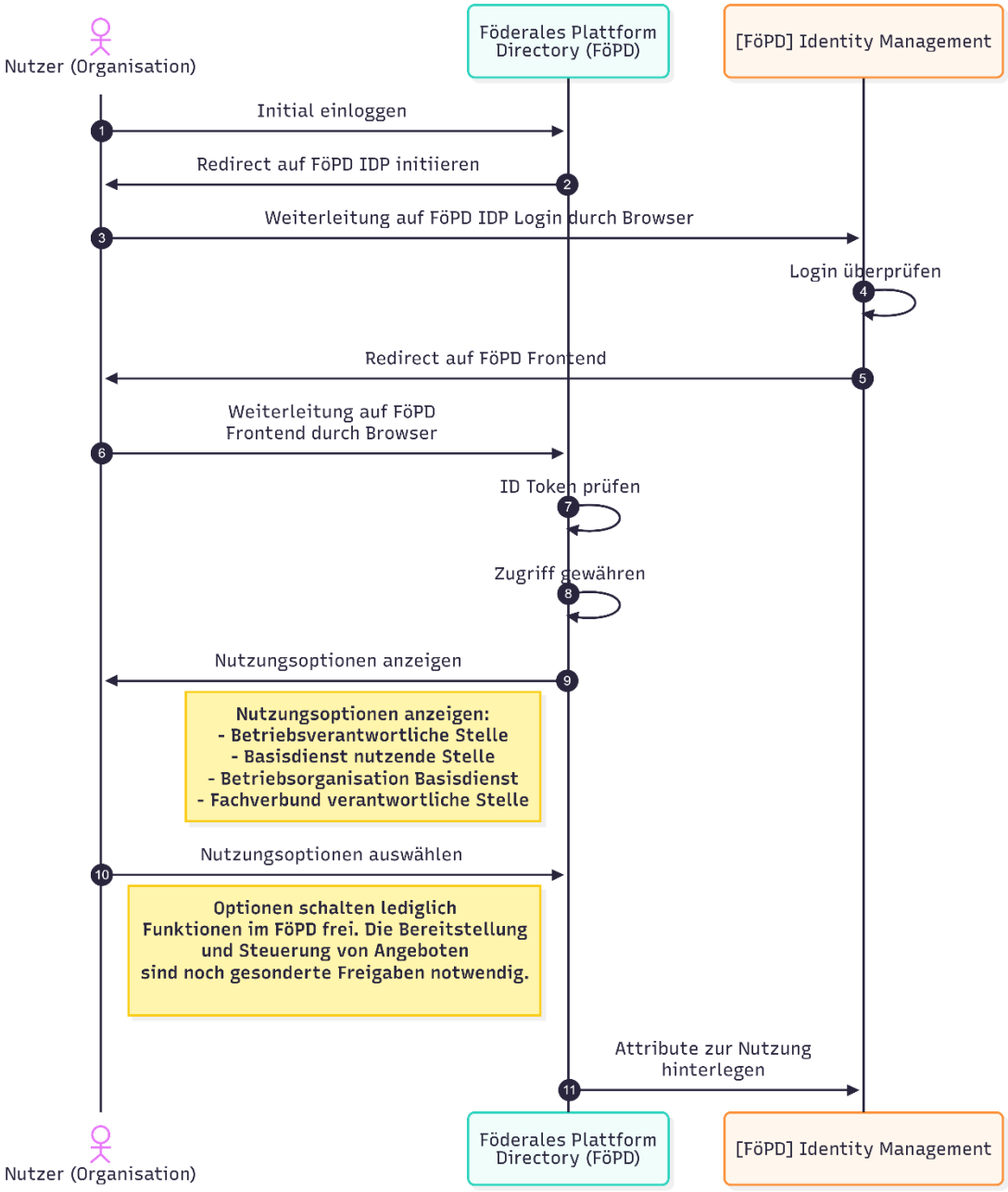


Abbildung 22: Sequenzdiagramm für den Prozess Organisation registrieren“ - Teil 2

4.4.2.2 Prozess „Basisdienst Angebote ermitteln und nutzen“

Der Prozess „Basisdienst Angebote ermitteln und nutzen“ ist keiner übergeordneten Prozessgruppe zugeordnet und unterstützt die Wertströme „APIs von Basisdiensten integrieren“ und „Angebote von Basisdiensten nutzen“. Er umfasst die Ermittlung der für eine Organisation verfügbaren Basisdienst-Angebote (APIs und Frontend-Angebote) sowie die Überführung dieser Angebote in die aktive Nutzung durch die Organisation. Ob ein Basisdienst-Angebot für eine

Organisation verfügbar ist, ergibt sich aus den vorhandenen Eigenschaften der Organisation in Verbindung mit der für das Angebot hinterlegten Policy; eine zusätzliche Beantragung von API-spezifischen Berechtigungen entfällt. Ein Angebot in aktiver Nutzung bildet die Grundlage dafür, dass die Organisation die zugehörigen Nutzungsoptionen wahrnehmen kann, etwa eine Software für eine API anzulegen, einer Software eine M2M-Autorisierung zu erteilen oder Nutzern den Zugriff auf ein Frontend-Angebot zu autorisieren.

Relevante Use Cases:

Tabelle 21: Use Cases des Prozesses „Basisdienst Angebote ermitteln und nutzen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Kataloginformationen bereitstellen	Föderales Plattform Directory (FöPD)	FöPD Nutzer	Anzeige der katalogisierten Basisdienst-Angebote (APIs und Frontend-Angebote) mit ihren Nutzungsbedingungen und der für die Organisation aktuell bestehenden Nutzungsberechtigung; Entgegennahme der Auswahl der in die aktive Nutzung zu überführenden Angebote.
Workflow Management bereitstellen	Föderales Plattform Directory (FöPD)	FöPD Nutzer	Unterstützung des Auswahl- und Übernahmeprozesses: Speicherung und revisionssichere Protokollierung der Nutzungsauswahl sowie der Zustimmung zu den Nutzungsbedingungen, Aktualisierung der internen Berechtigungen zur Nutzung der ausgewählten Angebote und Benachrichtigung des Organisationsaccounts sowie einer ggf. hinterlegten E-Mail-Adresse über die erfolgte Freigabe.
Policy prüfen	Zentrale Policy Infrastruktur	Föderales Plattform Directory (FöPD)	Auswertung einer Policy-Abfrage des FöPD und Rückgabe einer Policy-Entscheidung darüber, für welche Basisdienst-Angebote eine Organisation auf Basis ihrer Eigenschaften aktuell nutzungsberechtigt ist.

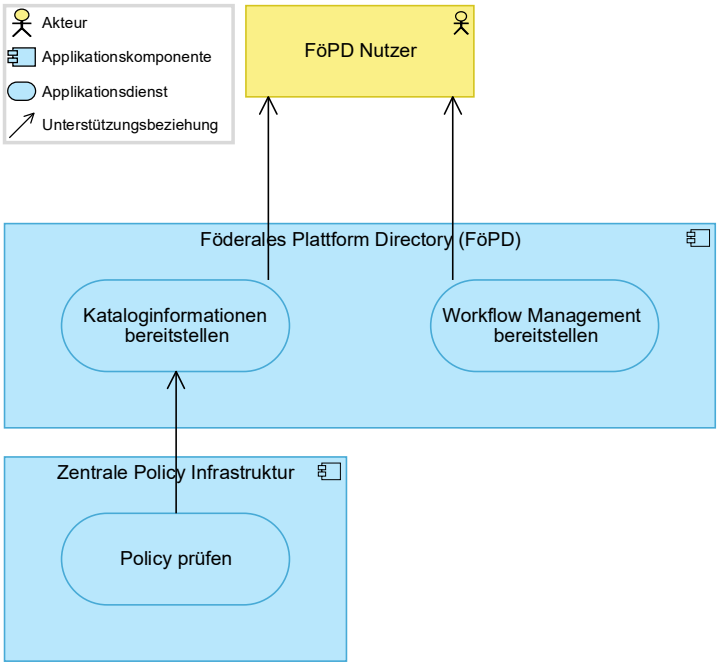


Abbildung 23: Use-Case-Nutzungsdiagramm für den Prozess „Basisdienst Angebote ermitteln und nutzen"

Informationsfluss im Prozessablauf:

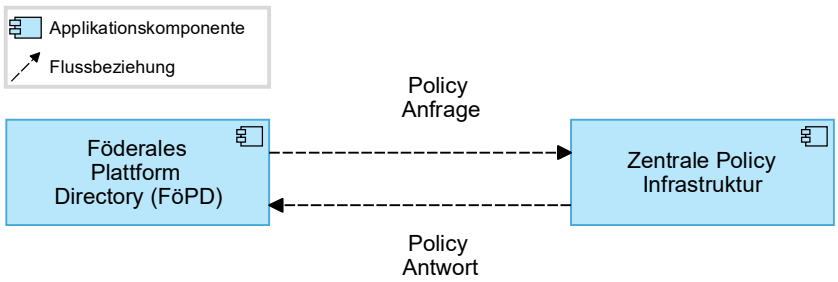


Abbildung 24: Informationsflussdiagramm für den Prozess „Basisdienst Angebote ermitteln und nutzen"

Ablaufbeschreibung

Der Prozess wird durch einen FöPD-Nutzer einer Organisation initiiert, der den Katalog der Basisdienst-Angebote im Self-Service-Portal des FöPD öffnet. Das FöPD prüft daraufhin bei der Zentralen Policy Infrastruktur, für welche katalogisierten Basisdienst-Angebote die Organisation auf Basis ihrer aktuell hinterlegten Eigenschaften nutzungsberechtigt ist. Die Zentrale Policy Infrastruktur wertet die zugehörigen Policies aus und liefert das Berechtigungsergebnis an das FöPD zurück. Anschließend zeigt das FöPD dem Nutzer die katalogisierten Basisdienst-Angebote zusammen mit der jeweils aktuellen Nutzungsberechtigung der Organisation an. Ob diese Berechtigungsabfrage aus Performancegründen über einen lokalen Policy Decision Point beim FöPD erfolgen sollte, ist architekturell noch nicht abschließend entschieden.

Sofern die Organisation für bestimmte Angebote nicht über die notwendigen Eigenschaften verfügt, sind diese ergänzend zu hinterlegen. Sie können entweder im FöPD über den Prozess „Organisationseigenschaften beantragen" beantragt oder bei der zuständigen Attribute Authority erfasst werden. Über die Zulässigkeit der jeweiligen Eigenschaft entscheidet die freigebende Stelle im FöPD beziehungsweise die zuständige Attribute Authority.

Aus den nutzungsberechtigten Angeboten wählt der Nutzer diejenigen aus, die seine Organisation aktiv in die Nutzung überführen möchte. Die im FöPD verfügbaren Nutzungsoptionen unterscheiden sich nach Art des Angebots. Bei API-Angeboten kann eine betriebsverantwortliche Stelle beispielsweise eine eigene Software anlegen, eine Basisdienst-nutzende Stelle einer Software eine M2M-Autorisierung erteilen oder eine Organisation Nutzer autorisieren, in ihrem Namen Software anzulegen oder einer Software eine M2M-Autorisierung zu geben. Bei Frontend-Angeboten wie Fachportalen können Nutzer der Organisation für die Nutzung autorisiert werden. Sofern für ein ausgewähltes Angebot Nutzungsbedingungen vorliegen, akzeptiert der Nutzer diese im selben Schritt.

Das FöPD speichert die Nutzungsauswahl sowie die Zustimmung zu den Nutzungsbedingungen und protokolliert sie revisionssicher. Anschließend aktualisiert es die internen Berechtigungen der Organisation zur Nutzung der ausgewählten Basisdienst-Angebote, sodass die zugehörigen Folgeprozesse, etwa „Software anlegen" oder „Auf Basisdienstfrontends zugreifen", für die Organisation freigeschaltet werden. Abschließend benachrichtigt das FöPD den Organisationsaccount sowie eine ggf. hinterlegte E-Mail-Adresse über die erfolgte Freigabe.

Ob die Organisation weiterhin nutzungsberechtigt bleibt, wird laufend über die Policy Infrastruktur geprüft. Änderungen können sich insbesondere durch geänderte Policies oder durch den Entzug von Organisationseigenschaften, Autorisierungen oder Delegationen ergeben.

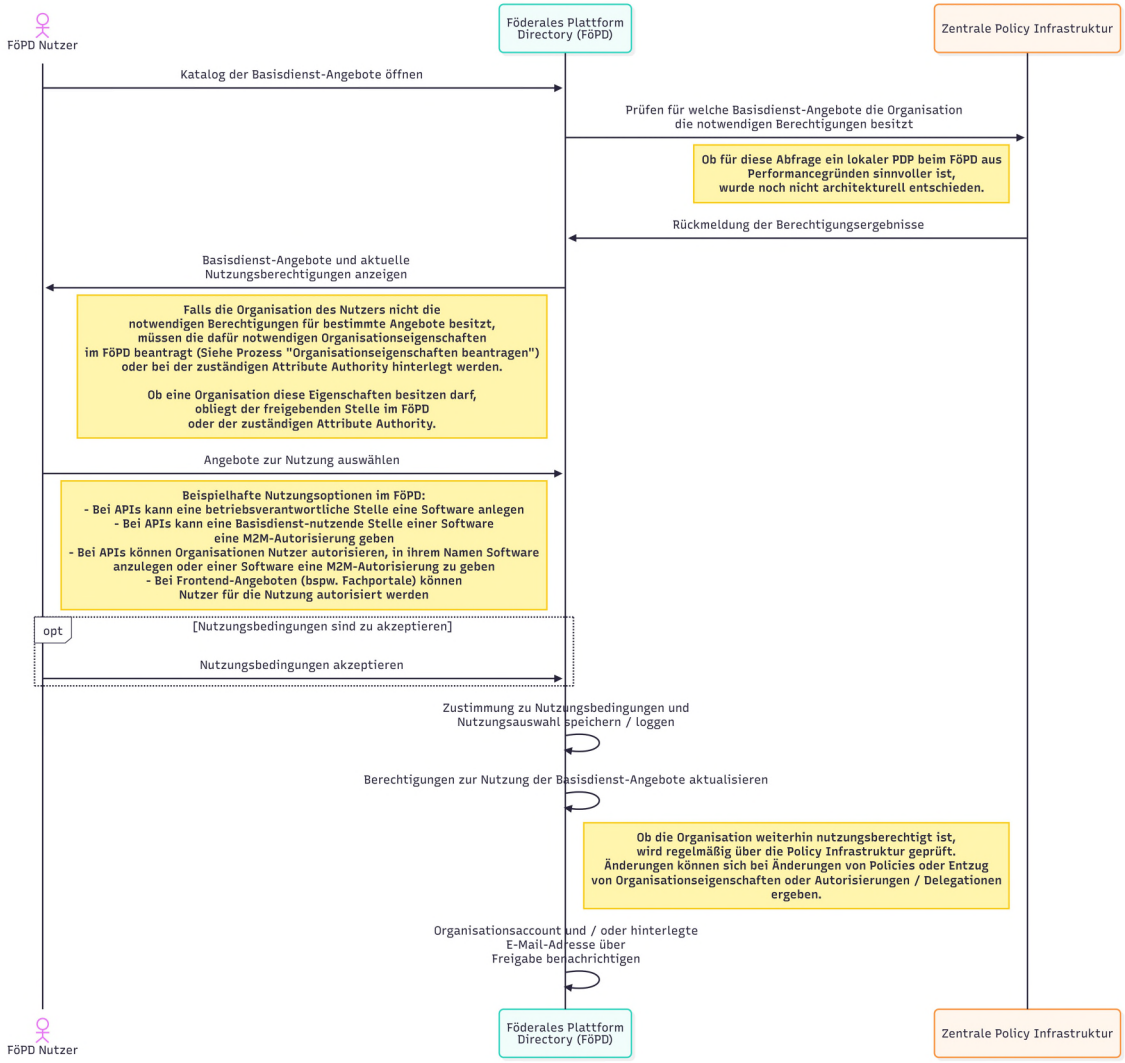


Abbildung 25: Sequenzdiagramm für den Prozess „Basisdienst Angebote ermitteln und nutzen“

4.4.2.3 Prozess „Organisationseigenschaften beantragen“

Der Prozess „Organisationseigenschaften beantragen“ gehört zur Prozessgruppe der zentralen FöPD-Beantragungsprozesse und unterstützt die Wertströme „APIs von Basisdiensten integrieren“, „APIs von Basisdiensten bereitstellen“ sowie „Angebote von Basisdiensten nutzen“. Er umfasst die Beantragung von Organisationsberechtigungen durch einen beim Föderalen

Plattform Directory registrierten Basisdienstnutzer, etwa die Zuweisung bestimmter Organisationsrollen oder -attribute, die für die Nutzung von Plattformangeboten oder die Wahrnehmung spezifischer Funktionen im FöPD erforderlich sind.

Relevante Use Cases:

Tabelle 22: Use Cases des Prozesses „Organisationseigenschaften beantragen"

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Organisationseigenschaften verwalten	Föderales Plattform Directory (FöPD)	Betriebsverantwortliche Stelle – API Consumer	Anzeige beantragbarer APIs mit Berechtigungsoptionen und Voraussetzungen; Entgegennahme von Berechtigungsanträgen und Aktualisierung der Organisationsattribute nach erfolgter Freigabe.
Workflow Management bereitstellen	Föderales Plattform Directory (FöPD)	Betriebsverantwortliche Stelle – API Consumer, Freigebende Stelle für API-Zugriffe	Unterstützung des Antrags- und Freigabeprozesses: Anlage und Verwaltung von Vorgängen, Benachrichtigung aller Beteiligten sowie Weiterleitung von Freigabeanfragen an die freigebende Stelle.
Identitätsdaten bereitstellen	FöPD Identity Provider	Zentrale Policy Infrastruktur	Bereitstellung aktueller Organisationsattribute für die Zentrale Policy Infrastruktur als Grundlage für den Attribute Store.
Identitätsdaten aktualisieren	FöPD Identity Provider	Föderales Plattform Directory (FöPD)	Speicherung und Aktualisierung der Organisationsattribute im Identity Provider nach erfolgter Freigabe.
Attribute Store verwalten	Zentrale Policy Infrastruktur	FöPD Identity Provider	Verwaltung der für Berechtigungsregeln relevanten Attribute aus FöPD und FöPD Identity Provider als normative Grundlage für Policy-Entscheidungen zur Laufzeit.

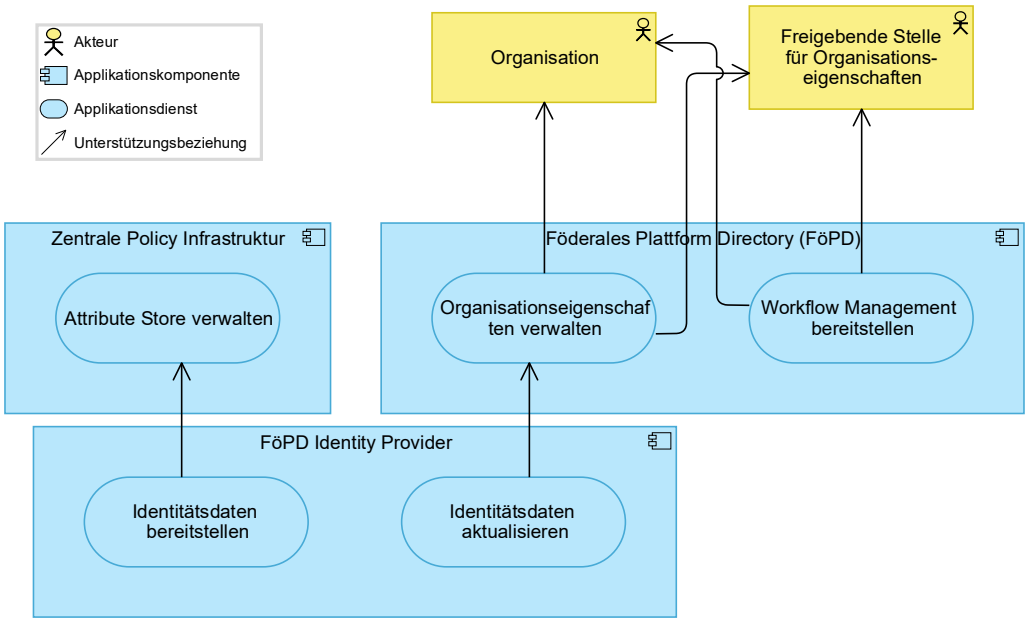


Abbildung 26: Use-Case-Nutzungsdiagramm für den Prozess „Organisationseigenschaften beantragen“

Informationsfluss im Prozessablauf:

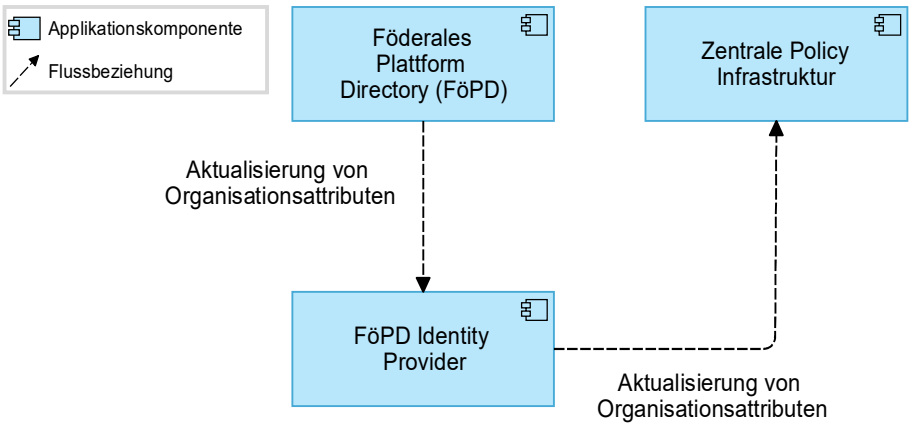


Abbildung 27: Informationsflussdiagramm für den Prozess „Organisationseigenschaften beantragen“

Ablaufbeschreibung

Der Prozess wird durch einen Nutzer einer Organisation initiiert, der über das Self-Service-Portal des FöPD die beantragbaren Attribute einsieht. Die Annahme dabei ist, dass der Nutzer weiß, für welchen Zweck er das jeweilige Attribut benötigt. Der Nutzer stellt den Antrag auf das gewünschte Attribut. Das FöPD legt daraufhin einen Vorgang an und bestätigt dem Nutzer die Vorgangsanlage. Der Organisationsaccount sowie ggf. eine hinterlegte E-Mail-Adresse werden über die Vorgangsanlage benachrichtigt.

Das FöPD benachrichtigt anschließend die zuständige freigebende Stelle über den Antrag. Diese prüft den Antrag, was mehrere Tage oder Wochen in Anspruch nehmen kann. Die Art der Prüfung ist fachspezifisch und kann Hintergrundrecherchen zur Organisation sowie Gesetzespüfungen beinhalten. Bei positiver Entscheidung gibt die freigebende Stelle den Antrag frei.

Nach erfolgter Freigabe aktualisiert das FöPD die Attribute der Organisation im FöPD Identity Provider. Abschließend benachrichtigt das FöPD den Organisationsaccount sowie ggf. eine hinterlegte E-Mail-Adresse über die erfolgte Freigabe.

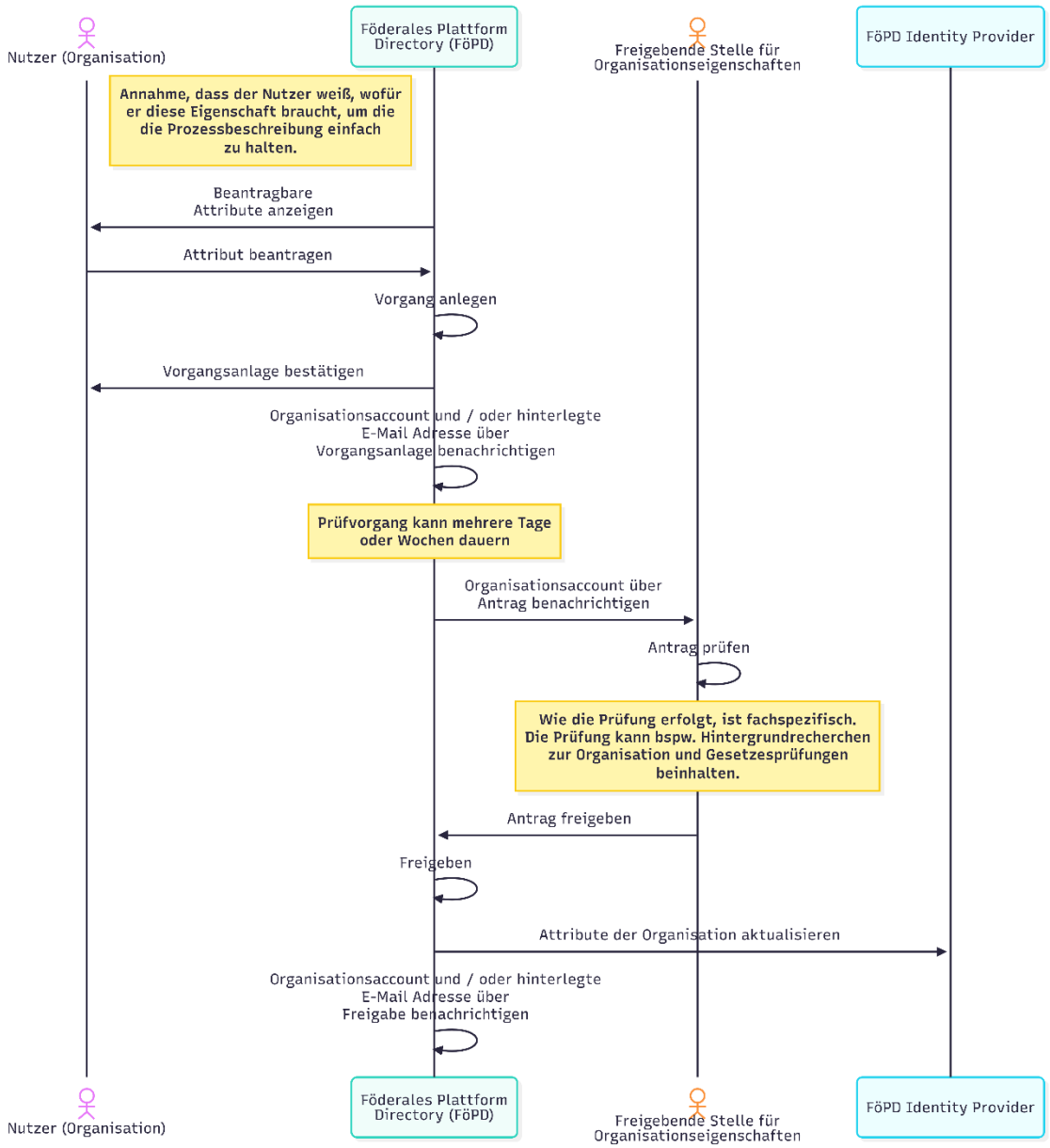


Abbildung 28: Sequenzdiagramm für den Prozess „Organisationseigenschaften beantragen“

4.4.2.4 Prozess „Software anlegen“

Der Prozess „Software anlegen“ gehört zur Prozessgruppe „API-Consumer und deren Berechtigungen verwalten“ und unterstützt den Wertstrom „APIs von Basisdiensten integrieren“. Er umfasst die Anlage einer API-Consumer-Anwendung als Software im Föderalen Plattform Directory, inklusive der Zuweisung der benötigten API-Berechtigungen und optional einer M2M-Autorisierung durch eine Basisdienst nutzende Stelle. Die registrierte Software bildet die

Grundlage für die spätere Ausstellung von Software Statement Assertions und die Registrierung von API-Clients bei den jeweiligen Basisdiensten.

Relevante Use Cases:

Tabelle 23: Use Cases des Prozesses "Software anlegen"

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Software verwalten	Föderales Plattform Directory (FöPD)	Betriebsverantwortliche Stelle – API Consumer	Anlage einer Software im FöPD mit den gewünschten APIs, Berechtigungen, Metadaten und Public Keys als Grundlage für die Ausstellung von Software Statement Assertions.
M2M Client autorisieren	Föderales Plattform Directory (FöPD)	Basisdienst Nutzer	Optionale M2M-Autorisierung der Software durch einen Basisdienst Nutzer. Dabei werden die notwendigen Organisationseigenschaften geprüft und die Autorisierung sowie zugehörige Attribute hinterlegt.
Software Informationen bereitstellen	Föderales Plattform Directory (FöPD)	Zentrale Policy Infrastruktur	Bereitstellung der relevanten Software-Informationen an die Zentrale Policy Infrastruktur zur Aktualisierung des Attribute Stores.
Software Statement Assertion ausstellen	Föderales Plattform Directory (FöPD)	Betriebsverantwortliche Stelle – API Consumer	Generierung des Software Statements und Signierung als Software Statement Assertion zum Abruf durch die betriebsverantwortliche Stelle.
Attribute Store verwalten	Zentrale Policy Infrastruktur	–	Aktualisierung des Attribute Stores mit den Software-Informationen des API-Consumers als Grundlage für Policy-Entscheidungen zur Laufzeit.

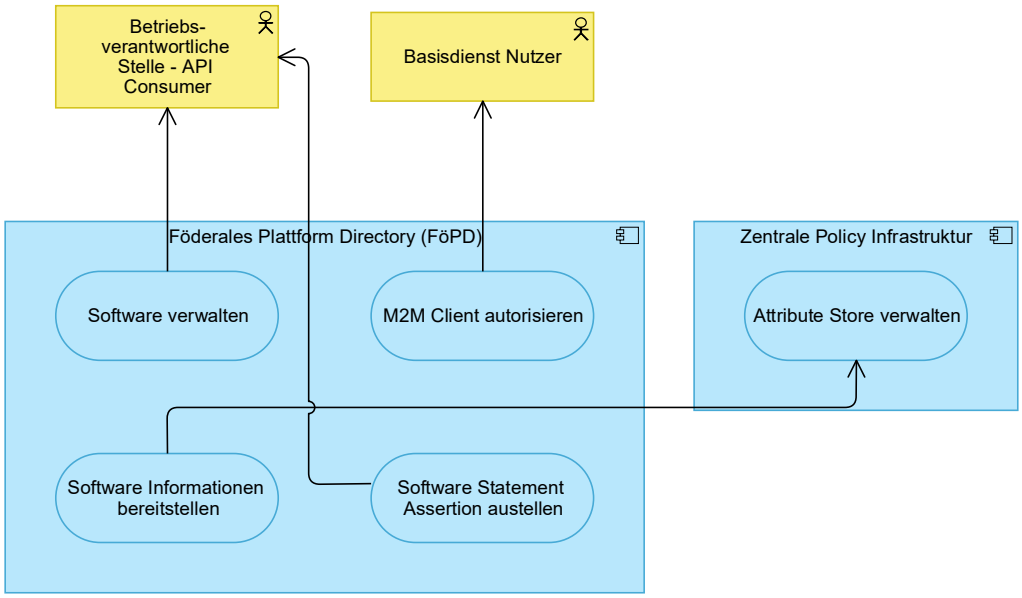


Abbildung 29: Use-Case-Nutzungsdiagramm für den Prozess „Software anlegen“

Informationsfluss im Prozessablauf:

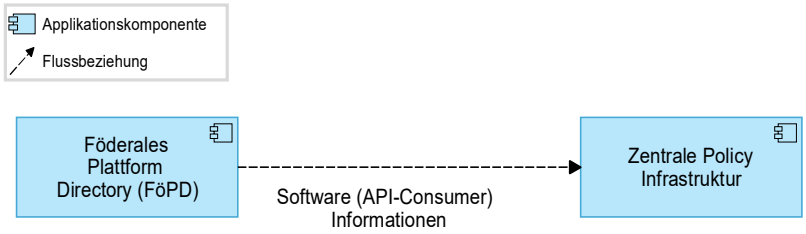


Abbildung 30: Informationsflussdiagramm für den Prozess „Software anlegen“

Ablaufbeschreibung

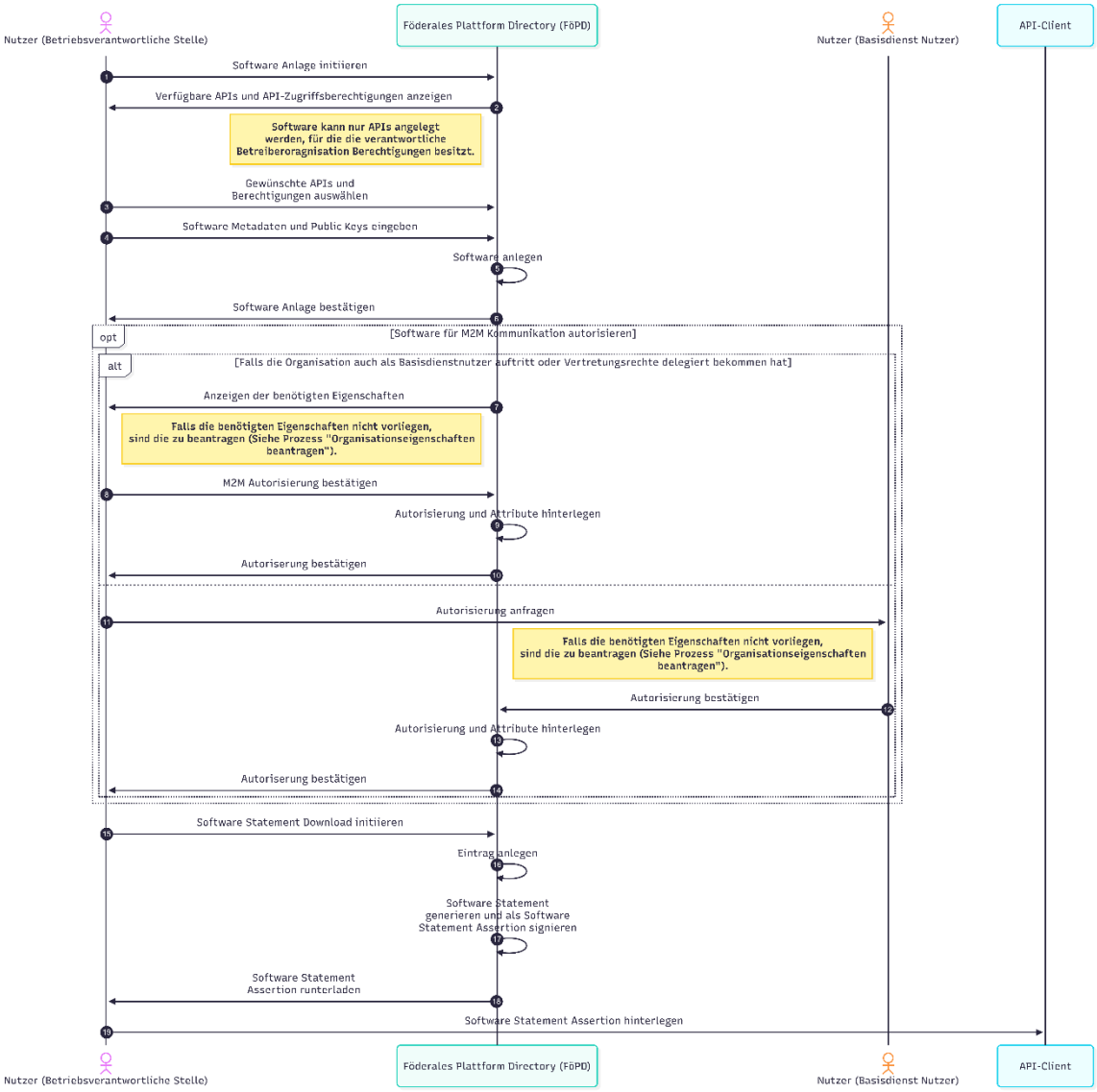


Abbildung 31: Sequenzdiagramm für den Prozess „Software anlegen“

Der Prozess wird durch einen Nutzer der betriebsverantwortlichen Stelle initiiert, der die Software-Anlage über das FöPD startet. Das FöPD zeigt die verfügbaren APIs und API-Zugriffsberechtigungen an. Dabei gilt, dass eine Software nur für APIs angelegt werden kann, für die die verantwortliche Betreiberorganisation bereits Berechtigungen besitzt. Der Nutzer wählt die gewünschten APIs und Berechtigungen aus und gibt anschließend die Software-Metadaten sowie die zugehörigen Public Keys ein. Das FöPD legt die Software an und bestätigt die Anlage.

Optional kann die Software im selben Schritt für M2M-Kommunikation autorisiert werden. Die Autorisierung erfolgt durch einen Basisdienst Nutzer, der entweder selbst der Organisation angehört oder Vertretungsrechte delegiert bekommen hat. Das FöPD zeigt die hierfür benötigten Eigenschaften an. Falls diese nicht vorliegen, sind sie zunächst über den Prozess „Organisationseigenschaften beantragen" zu beantragen. Nach Bestätigung hinterlegt das FöPD die Autorisierung und die zugehörigen Attribute.

Nach der Anlage initiiert der Nutzer den Download der Software Statement Assertion. Das FöPD legt einen Eintrag an, generiert das Software Statement und signiert es als Software Statement Assertion. Die SSA wird zum Download bereitgestellt und vom Nutzer heruntergeladen. Anschließend hinterlegt der Nutzer die SSA im API-Client. Hinweis: Im übergreifenden Informationsfluss ist darüber hinaus ein direkter API-Zugriff durch den API-Client oder eine Client Verwaltungssoftware auf das FöPD vorgesehen. Dieser automatisierte Pfad wird im Rahmen dieses Prozesses nicht im Detail beschrieben.

4.4.2.5 Prozess „API-Clients registrieren"

Der Prozess „API-Clients registrieren" gehört zur Prozessgruppe „API-Consumer und deren Berechtigungen verwalten" und unterstützt den Wertstrom „APIs von Basisdiensten integrieren". Er umfasst die automatische Registrierung von API-Clients bei allen relevanten Basisdiensten mittels Software Statement Assertions. Die Registrierung erfolgt über den Dynamic Client Registration Mechanismus und schafft die technischen Voraussetzungen dafür, dass ein API Consumer Client anschließend Access Tokens bei den jeweiligen Basisdienst Authorization Servern abrufen kann.

Relevante Use Cases:

Tabelle 24: Use Cases des Prozesses „API-Clients registrieren"

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
API-Clients registrieren und verwalten	[Basisdienst] Authorization Server	API Consumer Client	Entgegennahme der Client Registration Anfrage mit Software Statement Assertion, Prüfung und Validierung der SSA, Anlage des Clients nach positiver Policy-Entscheidung sowie Bestätigung der Clientanlage.

Policy prüfen	[Basisdienst] Dezentrale Policy Infrastruktur	[Basisdienst] Authorization Server	Auswertung der Berechtigungsanfrage des Authorization Servers und Rückgabe einer Policy-Entscheidung darüber, ob der Client registriert werden darf.

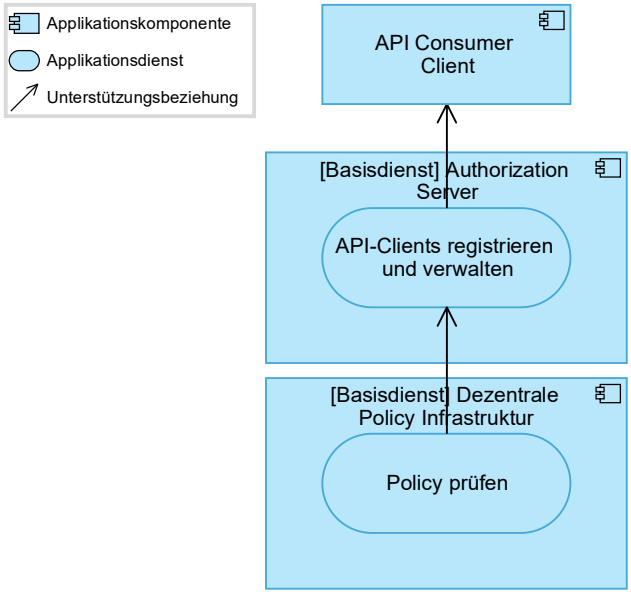


Abbildung 32: Use-Case-Nutzungsdiagramm für den Prozess „API-Clients registrieren“

Informationsfluss im Prozessablauf:

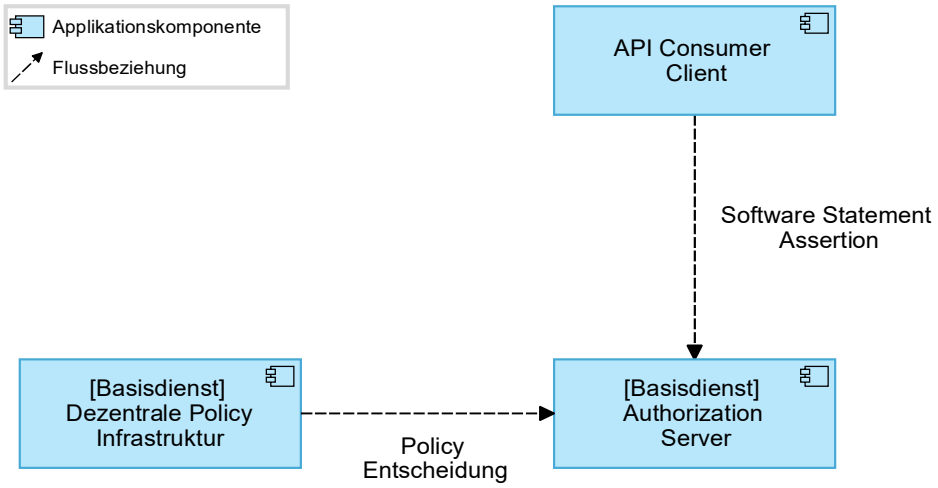


Abbildung 33: Informationsflussdiagramm für den Prozess „API-Clients registrieren“

Ablaufbeschreibung

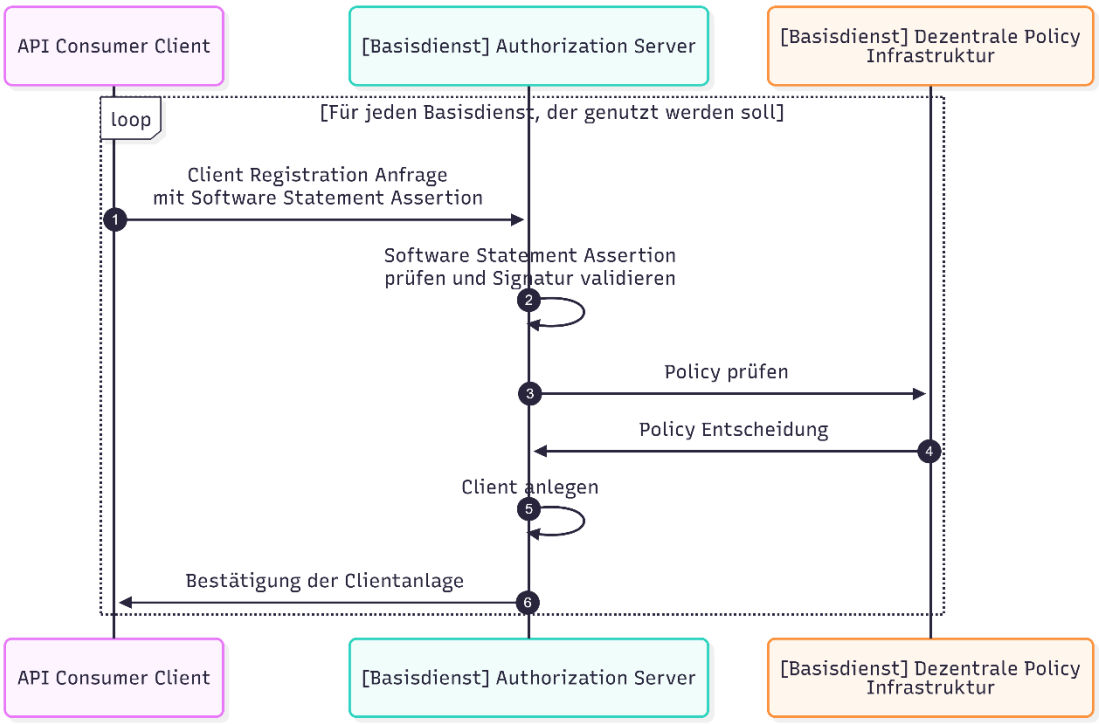


Abbildung 34: Sequenzdiagramm für den Prozess „API-Clients registrieren“

Der Prozess wird für jeden Basisdienst durchgeführt, der von der Software genutzt werden soll. Der API Consumer Client sendet eine Client Registration Anfrage inklusive der Software Statement Assertion an den Authorization Server des jeweiligen Basisdienstes. Der Authorization Server prüft die SSA und validiert deren Signatur. Anschließend leitet er eine Policy-Prüfung an die dezentrale Policy Infrastruktur des Basisdienstes weiter. Diese wertet die Anfrage anhand der hinterlegten Berechtigungsregeln aus und gibt eine Policy-Entscheidung zurück. Bei positiver Entscheidung legt der Authorization Server den Client an und bestätigt die Clientanlage gegenüber dem API Consumer Client. Dieser Ablauf wiederholt sich für jeden weiteren Basisdienst, der genutzt werden soll.

4.4.2.6 Prozess „Zugriff auf Basisdienst-Frontends ermöglichen“

Der Prozess „Zugriff auf Basisdienst-Frontends ermöglichen“ ist keiner übergeordneten Prozessgruppe zugeordnet und unterstützt den Wertstrom „Angebote von Basisdiensten nutzen“. Er umfasst die Authentifizierung von Basisdienstnutzern über das Föderale Plattform Directory sowie die Bereitstellung berechtigungsrelevanter Informationen und Attribute an den jeweiligen Basisdienst. Damit ermöglicht der Prozess den sicheren Zugriff auf Basisdienst-Frontendanwendungen durch natürliche Personen, die beim FöPD registriert sind.

Relevante Use Cases:

Tabelle 25: Use Cases des Prozesses „Zugriff auf Basisdienst-Frontends ermöglichen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Nutzer weiterleiten	Föderales Plattform Directory (FöPD)	Basisdienst Nutzer	Anzeige verfügbarer Basisdienst-Frontend-Angebote aus dem FöPD-Katalog und Weiterleitung des Nutzers zum gewählten Basisdienst-Frontend.
Frontend Funktionalitäten bereitstellen	Basisdienst	Basisdienst Nutzer	Entgegennahme der Zugriffsanfrage, Durchführung des OIDC Authorization Code Flows mit dem FöPD Identity Provider, Prüfung der Zugriffsberechtigung sowie Bereitstellung der Frontend-Anwendung bei positiver Entscheidung.
FöPD Nutzer authentifizieren	FöPD Identity Provider	Basisdienst	Authentifizierung des Basisdienst Nutzers im Rahmen des Authorization Code Flows. Bei vorhandener Session erfolgt die Authentifizierung als Silent

			SSO ohne Login-Dialog. Ausstellung von ID Token und Access Token nach erfolgreicher Authentifizierung.
Policy prüfen	[Basisdienst] Dezentrale Policy Infrastruktur	Basisdienst	Auswertung der Zugriffsprüfungsanfrage des Basisdienstes anhand der Nutzerattribute aus dem ID Token und Rückgabe einer Zugriffsentscheidung (PERMIT oder DENY).

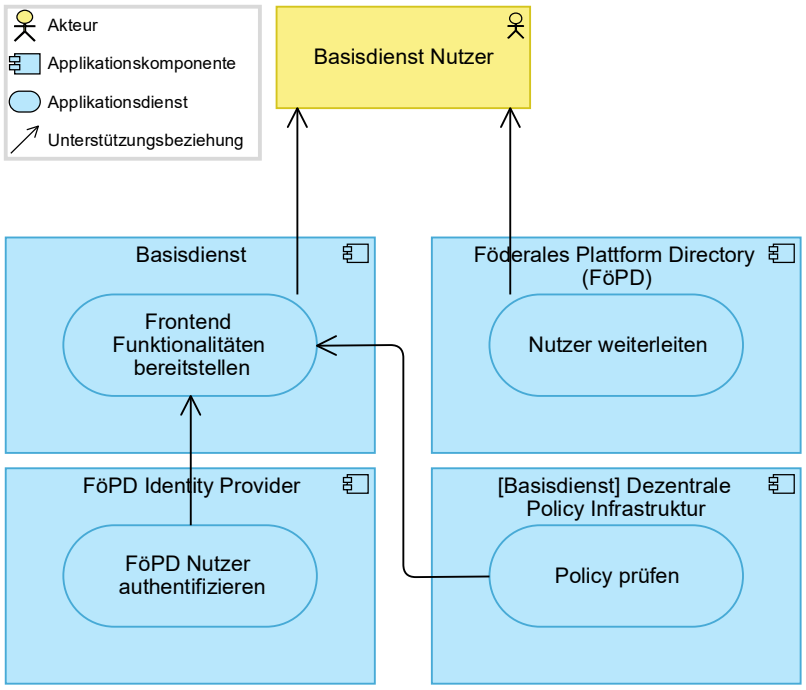


Abbildung 35: Use-Case-Nutzungsdiagramm für den Prozess „Zugriff auf Basisdienst-Frontends ermöglichen“

Informationsfluss im Prozessablauf:

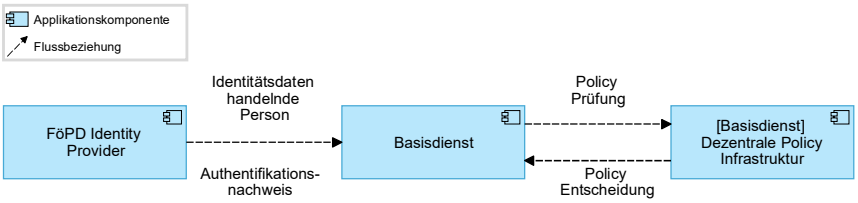


Abbildung 36: Informationsflussdiagramm für den Prozess „Zugriff auf Basisdienst-Frontends ermöglichen“

Ablaufbeschreibung

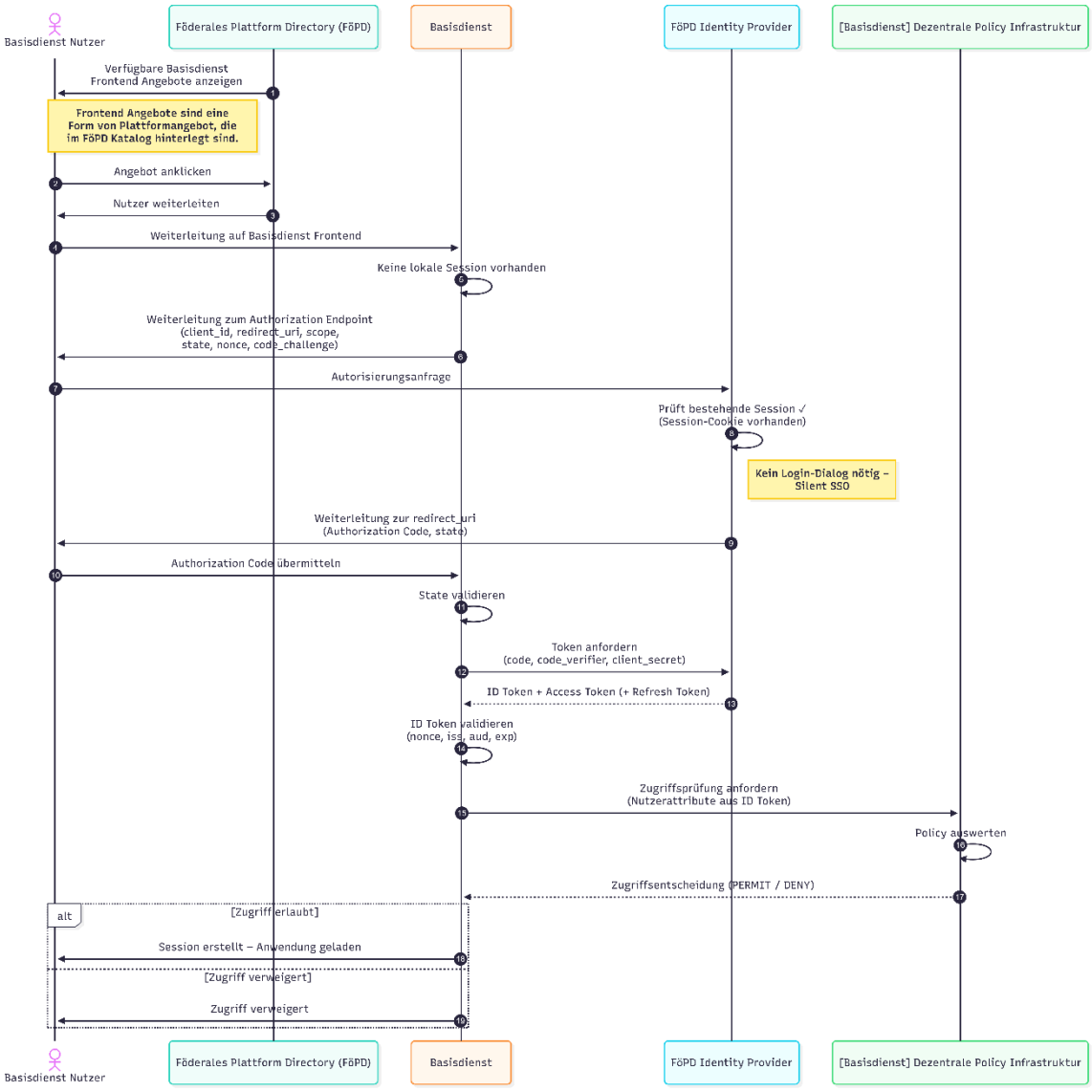


Abbildung 37: Sequenzdiagramm für den Prozess „Zugriff auf Basisdienst-Frontends ermöglichen“

Der Prozess beginnt damit, dass das FöPD dem Basisdienst Nutzer die verfügbaren Basisdienst-Frontend-Angebote aus dem FöPD-Katalog anzeigt. Frontend-Angebote sind eine Form von Plattformangebot, die im FöPD-Katalog hinterlegt sind. Der Nutzer wählt ein Angebot aus, woraufhin das FöPD ihn zum entsprechenden Basisdienst-Frontend weiterleitet.

Das Basisdienst-Frontend stellt fest, dass keine lokale Session vorhanden ist, und leitet den Nutzer zum Authorization Endpoint des FöPD Identity Providers weiter. Die Weiterleitung enthält die Parameter client_id, redirect_uri, scope, state, nonce und code_challenge. Der FöPD

Identity Provider prüft, ob eine bestehende Session vorliegt. Ist ein Session-Cookie vorhanden, erfolgt die Authentifizierung als Silent SSO ohne Login-Dialog. Der FöPD Identity Provider leitet den Nutzer anschließend mit Authorization Code und state zur redirect_uri des Basisdienstes zurück.

Der Basisdienst übermittelt den Authorization Code an den FöPD Identity Provider, validiert den state und fordert Token an. Der FöPD Identity Provider gibt ID Token, Access Token und optional Refresh Token zurück. Der Basisdienst validiert den ID Token anhand von nonce, iss, aud und exp.

Anschließend fordert der Basisdienst bei der dezentralen Policy Infrastruktur eine Zugriffsprüfung an und übermittelt dabei die Nutzerattribute aus dem ID Token. Die Policy Infrastruktur wertet die Anfrage aus und gibt eine Zugriffsentscheidung zurück. Bei PERMIT erstellt der Basisdienst eine Session und lädt die Anwendung für den Nutzer. Bei DENY wird dem Nutzer der Zugriff verweigert.

4.4.2.7 Prozess „Nutzerautorisierung erfassen“

Der Prozess „Nutzerautorisierung erfassen“ gehört zur Prozessgruppe „Basisdienst API nutzen“ und unterstützt den Wertstrom „Angebote von Basisdiensten nutzen“. Er umfasst die Erfassung zeitlich begrenzter Nutzerautorisierungen für API-Zugriffe von Anwendungen zur Laufzeit. Dabei autorisiert ein Basisdienstnutzer explizit, dass eine Anwendung in seinem Namen auf eine Basisdienst-API zugreifen darf. Das Ergebnis des Prozesses ist ein Access Token mit Nutzerautorisierung, der den weiteren API-Zugriff ermöglicht.

Relevante Use Cases:

Tabelle 26: Use Cases des Prozesses „Nutzerautorisierung erfassen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Nutzerautorisierung erfassen	Authorization Server für Nutzerzustimmung	Basisdienst Nutzer	Erfassung der expliziten Nutzereinwilligung für den Zugriff des API Consumer Clients auf die benötigten Basisdienst-APIs. Dem Nutzer werden Zweck und Umfang der Einwilligung in nicht-technischer Sprache dargestellt, inklusive aller Basisdienste, für die der Client autorisiert werden soll.

API-unspezifischen Access Token ausgeben	Authorization Server für Nutzerzustimmung	API Consumer Client	Ausstellung eines DPoP-gebundenen Opaque Access Tokens nach erfolgreicher Nutzereinwilligung. Der Token enthält die erfasste Nutzerautorisierung, ist jedoch noch nicht an einen spezifischen Basisdienst gebunden.
Natürliche Personen authentifizieren	BundID	Authorization Server für Nutzerzustimmung	Authentifizierung natürlicher Personen im Rahmen des Authorization Code Flows als Grundlage für die Erfassung der Nutzerautorisierung.
Juristische Personen authentifizieren	Mein Unternehmenskonto (MUK)	Authorization Server für Nutzerzustimmung	Authentifizierung juristischer Personen im Rahmen des Authorization Code Flows als Grundlage für die Erfassung der Nutzerautorisierung.
Natürliche oder juristische Person authentifizieren	Externe Identity Provider	Authorization Server für Nutzerzustimmung	Authentifizierung natürlicher oder juristischer Personen über externe föderale Identity Provider als Grundlage für die Erfassung der Nutzerautorisierung.

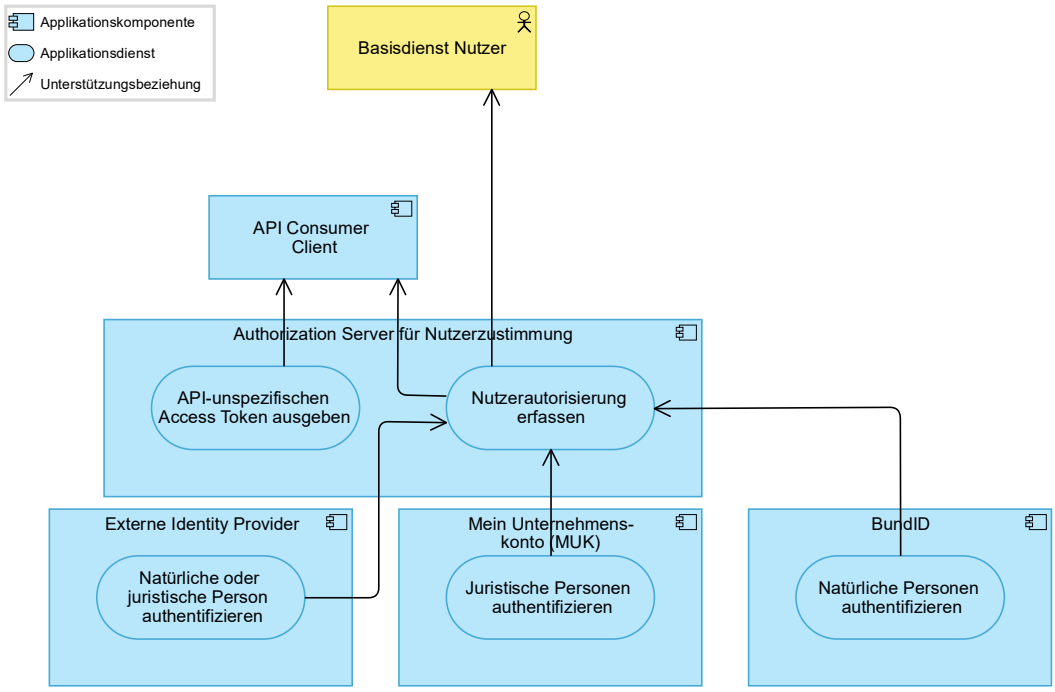


Abbildung 38: Use-Case-Nutzungsdiagramm für den Prozess „Nutzerautorisierung erfassen“

Informationsfluss im Prozessablauf:

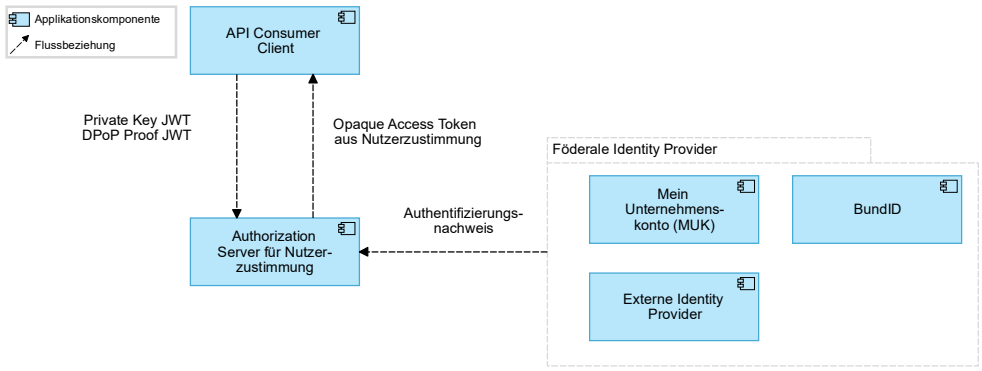


Abbildung 39: Informationsflussdiagramm für den Prozess „Nutzerautorisierung erfassen“

Ablaufbeschreibung

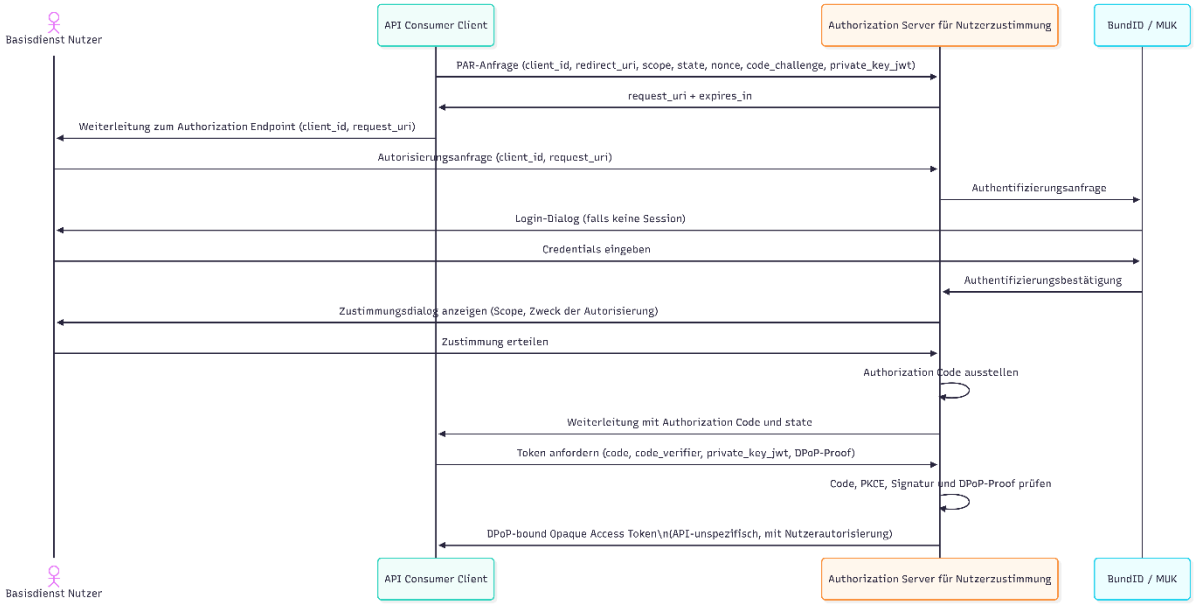


Abbildung 40: Sequenzdiagramm für den Prozess „Nutzerautorisierung erfassen“

Der Prozess wird durch den API Consumer Client initiiert, der den Basisdienst Nutzer zum Authorization Server für Nutzerzustimmung weiterleitet. Zunächst sendet der API Consumer Client eine PAR-Anfrage an den Authorization Server, die neben `client_id`, `redirect_uri`, `scope`, `state`, `nonce` und `code_challenge` auch das `private_key_jwt` zur Client-Authentifizierung enthält. Der Authorization Server gibt daraufhin eine `request_uri` mit Ablaufzeit zurück. Mit dieser

`request_uri` leitet der API Consumer Client den Basisdienst Nutzer zum Authorization Endpoint weiter.

Der Authorization Server fordert beim zuständigen Identity Provider eine Authentifizierung des Nutzers an. Je nach Nutzertyp übernimmt dies die BundID für natürliche Personen, Mein Unternehmenskonto für juristische Personen oder ein externer föderaler Identity Provider. Falls keine bestehende Session vorhanden ist, wird dem Nutzer ein Login-Dialog angezeigt. Nach erfolgreicher Authentifizierung zeigt der Authorization Server dem Nutzer einen Zustimmungsdialog. Darin wird in nicht-technischer Sprache dargestellt, für welche Zwecke die Einwilligung erteilt wird und für welche Basisdienste der API Consumer Client im Namen des Nutzers autorisiert werden soll. Der Nutzer erteilt seine Zustimmung.

Der Authorization Server stellt einen Authorization Code aus und leitet den Nutzer mit Code und `state` zur `redirect_uri` des API Consumer Clients zurück. Der Client fordert anschließend einen Token an und übermittelt dabei `code`, `code_verifier`, `private_key_jwt` sowie einen DPoP-Proof. Der Authorization Server prüft Code, PKCE, Signatur und DPoP-Proof und stellt bei positiver Prüfung einen DPoP-gebundenen Opaque Access Token aus. Dieser Token enthält die erfasste Nutzerautorisierung, ist jedoch noch API-unspezifisch. Der Austausch gegen einen basisdienstspezifischen Token erfolgt im nachgelagerten Prozess „Access Token abrufen“.

4.4.2.8 Prozess „Access Token abrufen“

Der Prozess „Access Token abrufen“ gehört zur Prozessgruppe „Basisdienst API nutzen“ und unterstützt den Wertstrom „Angebote von Basisdiensten nutzen“. Er umfasst den Abruf von Access Tokens durch eine Anwendung beim jeweiligen Basisdienst Authorization Server zum Zweck eines nachfolgenden API-Aufrufs. Der Prozess bildet zwei Varianten ab: Verfügt der API Consumer Client über eine M2M-Autorisierung, erfolgt der Token-Abruf über den Client Credentials Grant. Hat hingegen eine Autorisierung durch einen Nutzer zur Laufzeit stattgefunden, tauscht der Client das aus der Nutzerautorisierung erhaltene API-unspezifische Token mittels Token Exchange gegen ein basisdienst-spezifisches Token. In beiden Varianten authentifiziert sich der Client mittels `private_key_jwt`, weist den Besitz des DPoP-Schlüssels nach und erhält einen DPoP-gebundenen Opaque Access Token, der im Folgeschritt für den API-Aufruf verwendet wird.

Relevante Use Cases:

Tabelle 27: Use Cases des Prozesses „Access Token abrufen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
API Access Token für Client Credentials ausstellen	[Basisdienst] Authorization Server	API Consumer Client	Ausstellung eines DPoP-gebundenen Opaque Access Tokens auf Basis des Client Credentials Grant. Der Client authentifiziert sich mittels Private Key JWT und weist den Besitz des DPoP-Schlüssels nach. Der Authorization Server prüft die Berechtigung über die dezentrale Policy Infrastruktur und stellt bei positiver Entscheidung den Token aus.
API Access Token für Token Exchange ausstellen	[Basisdienst] Authorization Server	API Consumer Client	Ausstellung eines API-spezifischen, DPoP-gebundenen Opaque Access Tokens durch Eintausch eines API-unspezifischen Tokens aus einer vorherigen Nutzerautorisierung. Der Authorization Server löst das vorhandene Opaque Access Token über Token Introspection beim Authorization Server für Nutzerzustimmung auf, prüft die Berechtigung des Clients und stellt den basisdienst-spezifischen Token aus.
Policy prüfen	[Basisdienst] Dezentrale Policy Infrastruktur	[Basisdienst] Authorization Server	Prüfung der grobgranularen Berechtigung des API Consumer Clients für den Zugriff auf die angeforderte API. Die dezentrale Policy Infrastruktur trifft die Entscheidung und liefert das Ergebnis an den Authorization Server.
Opaque Access Token auflösen	Authorization Server für Nutzerzustimmung	[Basisdienst] Authorization Server	Auflösung des aus der Nutzerautorisierung stammenden Opaque Access Tokens mittels Token Introspection. Der Authorization Server für Nutzerzustimmung gibt ein JWS Access Token zurück, das die Nutzerautorisierung und die zugehörigen Claims enthält.

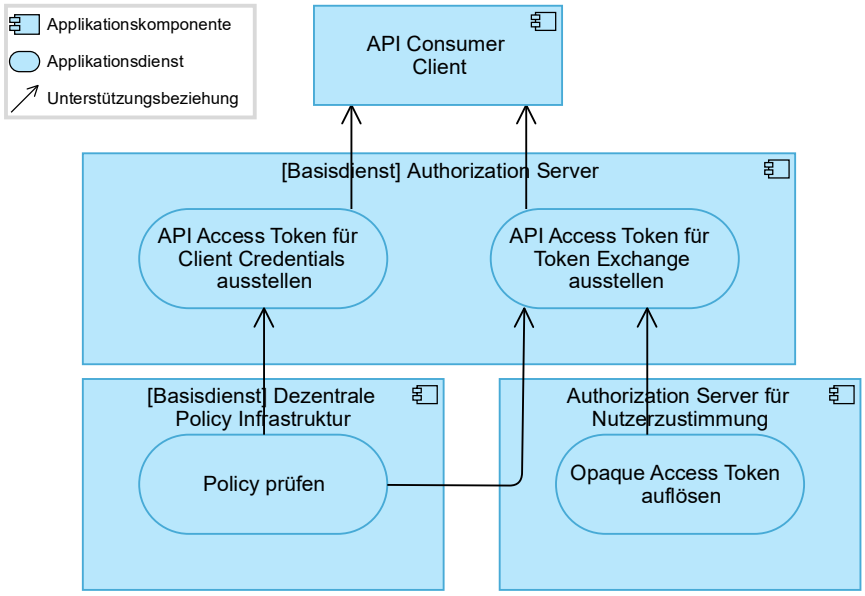


Abbildung 41: Use-Case-Nutzungsdiagramm für den Prozess „Access Token abrufen“

Informationsfluss im Prozessablauf:

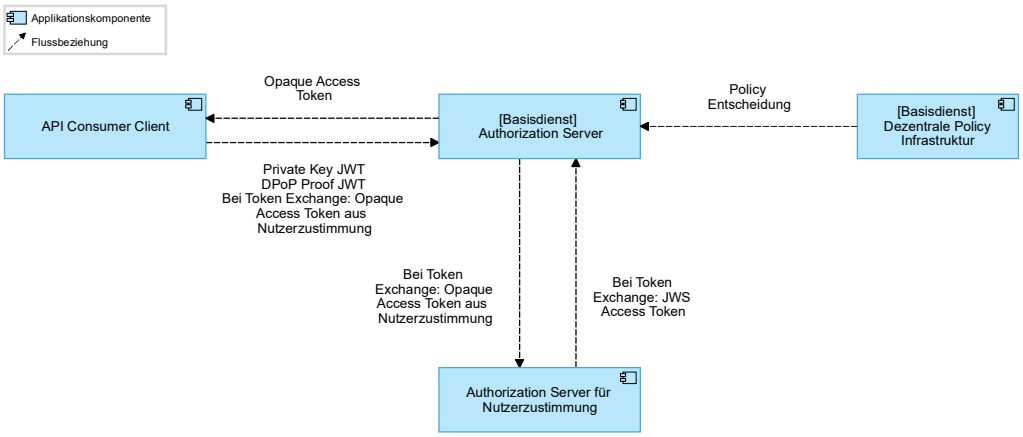


Abbildung 42: Informationsflussdiagramm für den Prozess „Access Token abrufen“

Ablaufbeschreibung

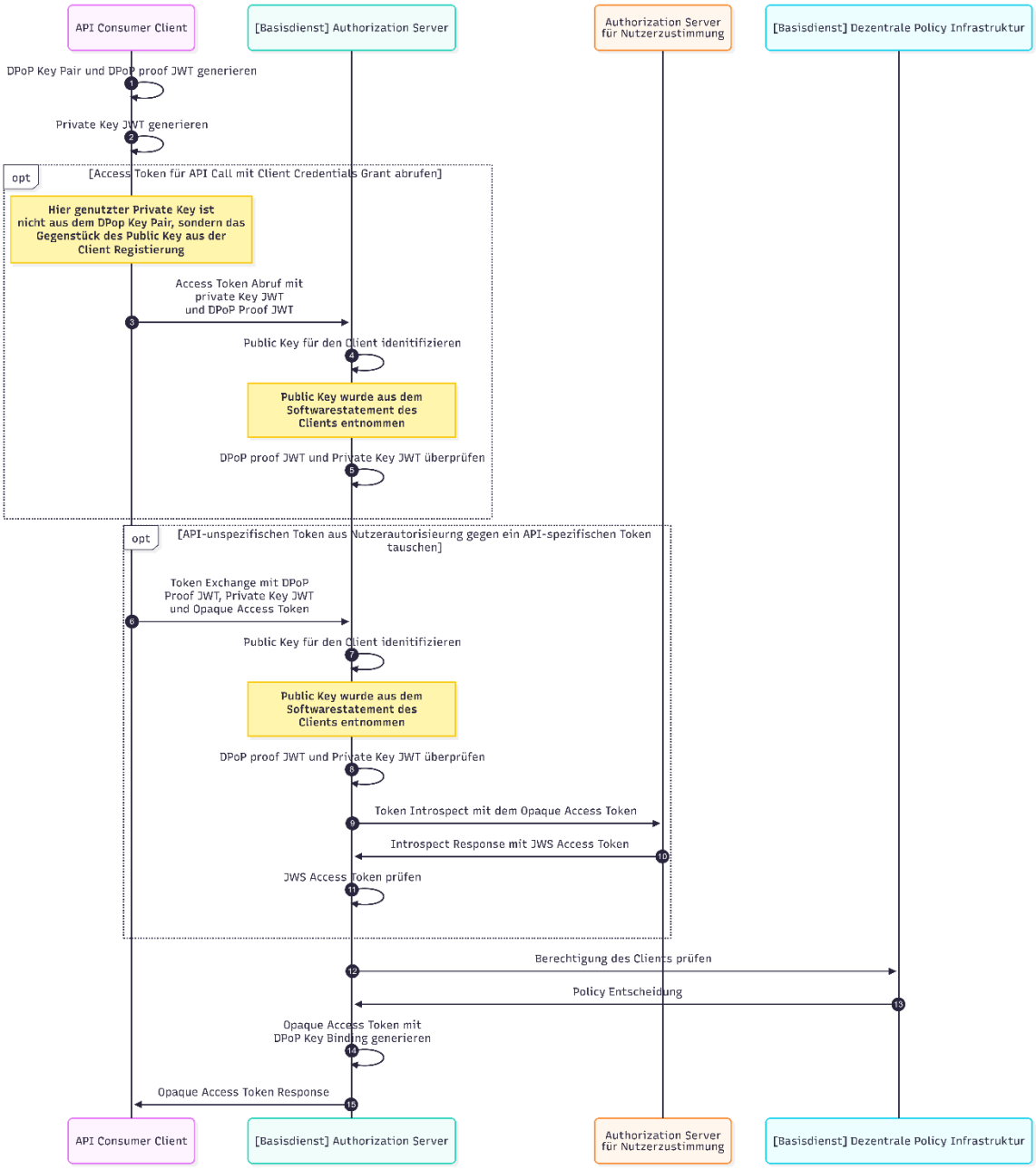


Abbildung 43: Sequenzdiagramm für den Prozess „Access Token abrufen"

Der API Consumer Client erzeugt zunächst ein DPoP Key Pair und erstellt daraus einen DPoP Proof JWT. Parallel generiert er ein `private_key_jwt` zur Client-Authentifizierung. Der hierfür verwendete private Schlüssel ist nicht identisch mit dem DPoP-Schlüssel, sondern das

Gegenstück des Public Keys, der im Rahmen der Client-Registrierung über das Software Statement hinterlegt wurde.

Verfügt der API Consumer Client über eine M2M-Autorisierung, sendet er eine Access-Token-Anfrage mit `private_key_jwt` und DPoP Proof JWT an den Basisdienst Authorization Server. Dieser identifiziert den zugehörigen Public Key aus dem Software Statement des Clients und überprüft sowohl den DPoP Proof JWT als auch das `private_key_jwt`.

Hat eine Nutzerautorisierung zur Laufzeit stattgefunden, sendet der API Consumer Client eine Token-Exchange-Anfrage an den Basisdienst Authorization Server. Neben `private_key_jwt` und DPoP Proof JWT übermittelt er das API-unspezifische Opaque Access Token aus dem vorgelagerten Nutzerautorisierungsprozess. Der Authorization Server identifiziert den Public Key des Clients aus dem Software Statement und prüft DPoP Proof JWT sowie `private_key_jwt`. Anschließend löst er das Opaque Access Token mittels Token Introspection beim Authorization Server für Nutzerzustimmung auf. Dieser gibt ein JWS Access Token zurück, das die Nutzerautorisierung und die zugehörigen Claims enthält. Der Basisdienst Authorization Server prüft das empfangene JWS Access Token.

Unabhängig von der gewählten Variante prüft der Basisdienst Authorization Server die Berechtigung des Clients über die dezentrale Policy Infrastruktur. Fällt die Policy-Entscheidung positiv aus, generiert der Authorization Server einen Opaque Access Token mit DPoP Key Binding und gibt diesen an den API Consumer Client zurück. Der ausgestellte Token ist damit kryptografisch an den DPoP-Schlüssel des Clients gebunden und kann im nachfolgen-den Prozess „API aufrufen“ verwendet werden.

4.4.2.9 Prozess „API aufrufen“

Der Prozess „API aufrufen“ gehört zur Prozessgruppe „Basisdienst API nutzen“ und unterstützt den Wertstrom „Angebote von Basisdiensten nutzen“. Er umfasst den Abruf einer Basisdienst-API mittels eines zuvor abgerufenen Access Tokens sowie die Prüfung des API-Aufrufs durch die dezentrale Infrastruktur des Basisdienstes. Die Prüfung beinhaltet die Token-Validierung durch das API-Gateway sowie eine Policy-Entscheidung durch die dezentrale Policy-Infrastruktur, bevor der Basisdienst den Zugriff auf die gewünschte API-Ressource gewährt.

Relevante Use Cases:

Tabelle 28: Use Cases des Prozesses „API aufrufen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
API-Anfragen vorprüfen	[Basisdienst] API-Gateway	API Consumer Client	Entgegennahme des API-Aufrufs mit Opaque Access Token und DPoP Header. Das API-Gateway löst das Opaque Access Token über Token Introspection beim Authorization Server auf, überprüft DPoP Header und Access Token und leitet die Berechtigungsprüfung über die dezentrale Policy Infrastruktur ein. Bei positivem Ergebnis wird der API-Aufruf mit dem JWS Access Token an den Basisdienst weitergeleitet.
JWS Token abrufen	[Basisdienst] API-Gateway	[Basisdienst] Authorization Server	Auflösung des vom API Consumer Client übermittelten Opaque Access Tokens mittels Token Introspection beim Basisdienst Authorization Server. Der Authorization Server prüft den Token-Eintrag und gibt ein JWS Access Token zurück, das vom API-Gateway für die weitere Verarbeitung und für künftige Requests gecacht wird.
API bereitstellen	Basisdienst	[Basisdienst] API-Gateway	Verarbeitung des vom API-Gateway weitergeleiteten API-Aufrufs mit JWS Access Token und Bereitstellung der angeforderten API-Ressource. Der Basisdienst gibt die API-Antwort an das API-Gateway zurück, das diese an den API Consumer Client weiterleitet.
Policy prüfen	[Basisdienst] Dezentrale Policy Infrastruktur	[Basisdienst] API-Gateway	Prüfung der Berechtigung des API Consumer Clients für den konkreten API-Zugriff auf Basis der im Access Token enthaltenen Informationen. Die dezentrale Policy Infrastruktur liefert das Policy-Prüfergebnis an das API-Gateway.
Opaque Access Token auflösen	[Basisdienst] Authorization Server	[Basisdienst] API-Gateway	Entgegennahme des Opaque Access Tokens via Token Introspection, Prüfung des Token-Eintrags und

			Generierung eines JWS Access Tokens, das die Token-Informationen in signierter Form enthält.

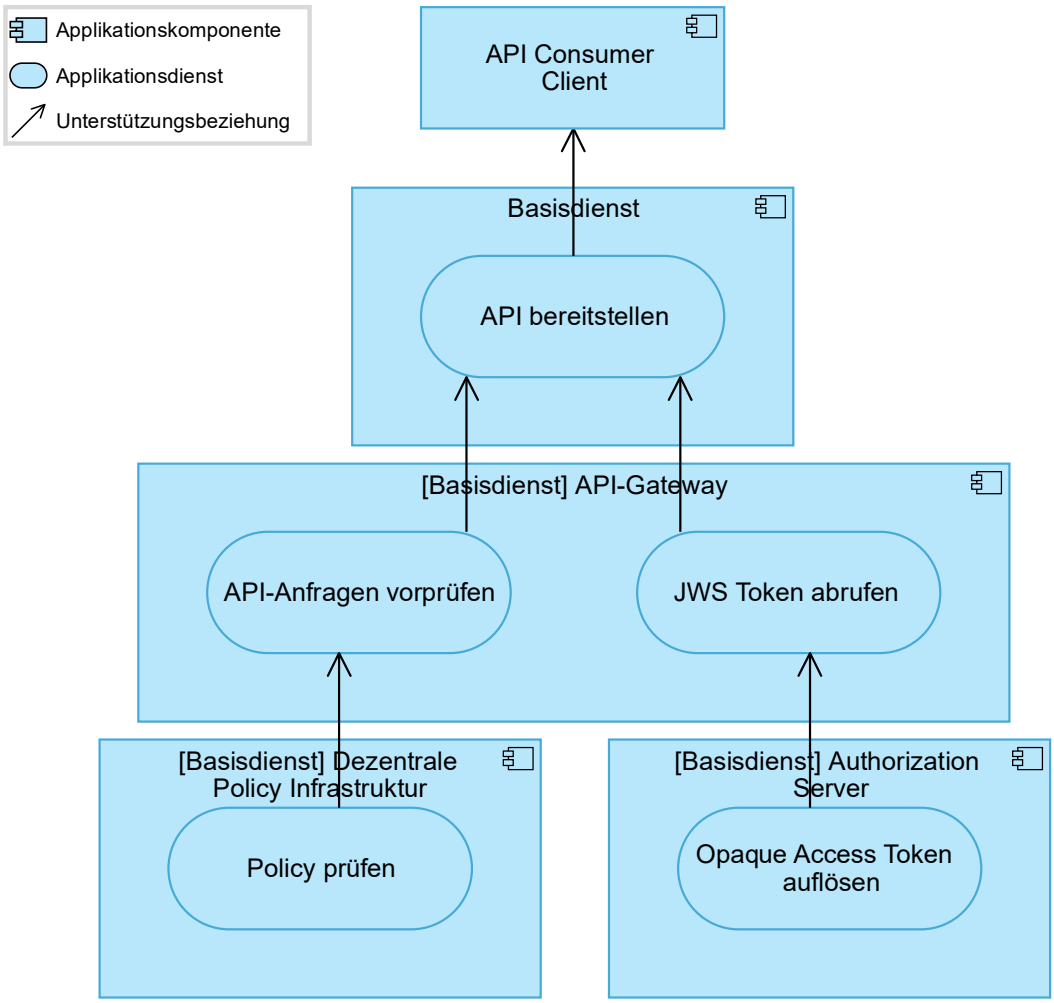


Abbildung 44: Use-Case-Nutzungsdiagramm für den Prozess „API aufrufen“

Informationsfluss im Prozessablauf:

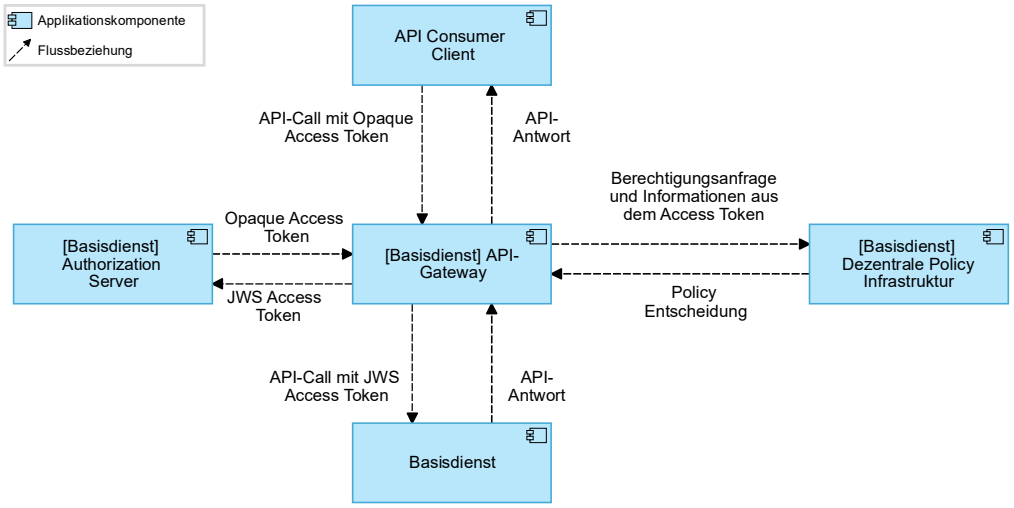


Abbildung 45: Informationsflussdiagramm für den Prozess „API aufrufen“

Ablaufbeschreibung

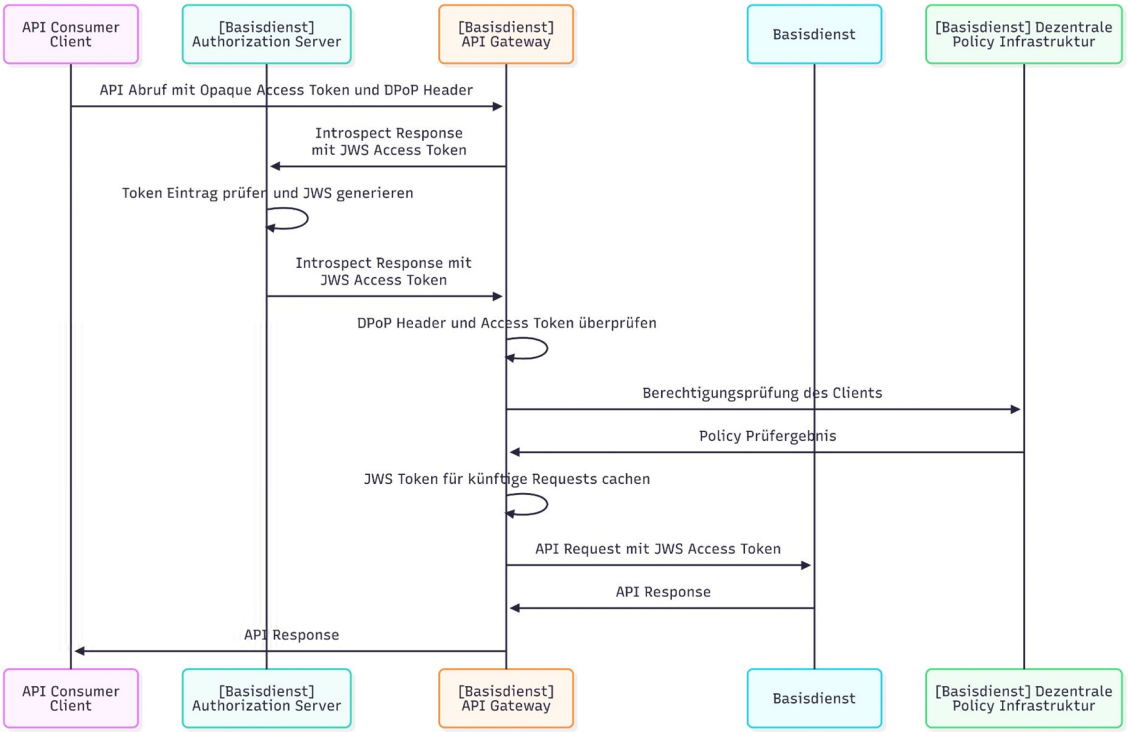


Abbildung 46: Sequenzdiagramm für den Prozess „API aufrufen“

Der API Consumer Client sendet einen API-Aufruf mit dem zuvor abgerufenen Opaque Access Token und einem DPoP Header an das Basisdienst API-Gateway. Das API-Gateway leitet das Opaque Access Token zur Auflösung an den Basisdienst Authorization Server weiter. Dieser prüft den Token-Eintrag, generiert ein JWS Access Token und gibt es per Introspect Response an das API-Gateway zurück.

Das API-Gateway überprüft den DPoP Header und das empfangene JWS Access Token. Anschließend leitet es eine Berechtigungsprüfung des Clients bei der dezentralen Policy Infrastruktur ein und übermittelt dabei Informationen aus dem Access Token. Die dezentrale Policy Infrastruktur trifft die Policy-Entscheidung und liefert das Prüfergebnis an das API-Gateway zurück.

Bei positivem Prüfergebnis cacht das API-Gateway das JWS Token für künftige Requests desselben Clients und leitet den API-Aufruf mit dem JWS Access Token an den Basisdienst weiter. Der Basisdienst verarbeitet die Anfrage und gibt eine API Response zurück, die das API-Gateway an den API Consumer Client weiterleitet.

4.4.2.10 Prozess „Plattformangebot festlegen“

Der Prozess „Plattformangebot festlegen“ gehört zur Prozessgruppe „Plattformangebote bereitstellen“ und unterstützt den Wertstrom „APIs von Basisdiensten bereitstellen“. Er umfasst die Katalogdefinition eines Plattformangebots durch eine berechtigte Stelle – etwa die Betriebsorganisation eines Basisdienstes oder eine fachverbundverantwortliche Stelle – sowie die anschließende Prüfung und vorläufige Freigabe durch die plattformverantwortliche Stelle. Das Ergebnis des Prozesses sind vollständig beschriebene Kataloginformationen als Grundlage für die spätere Berechtigungsmodell-Konfiguration und Veröffentlichung.

Relevante Use Cases:

Tabelle 29: Use Cases des Prozesses „Plattformangebot festlegen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Plattformangebot verwalten	Föderales Plattform Directory (FöPD)	Basisdienst Betreiber, Plattformverantwortliche Stelle	Anlage, Bearbeitung und Freigabe von Plattformangeboten. Der Basisdienst Betreiber beschreibt das Angebot fachlich und technisch über ein standardisiertes Formular. Die Plattformverantwortliche Stelle prüft das Angebot und gibt es frei.

Workflow Management bereitstellen	Föderales Plattform Directory (FöPD)	Basisdienst Betreiber, Plattformverantwortliche Stelle	Unterstützung des Antrags- und Freigabeprozesses: Benachrichtigung der Plattformverantwortlichen Stelle über neue Plattformangebote und Benachrichtigung des Organisationsaccounts über Anlage und Freigabe.
Kataloginformationen bereitstellen	Föderales Plattform Directory (FöPD)	Zentrale Policy Infrastruktur	Generierung der Katalogeinträge nach Freigabe des Plattformangebots und automatische Propagierung der Kataloginformationen zu Basisdienstangeboten an die Zentrale Policy Infrastruktur.
Policies verwalten	Zentrale Policy Infrastruktur	Föderales Plattform Directory (FöPD)	Entgegennahme und Speicherung der vom FöPD propagierten Kataloginformationen zu Basisdienstangeboten als Grundlage für die spätere Berechtigungssteuerung.

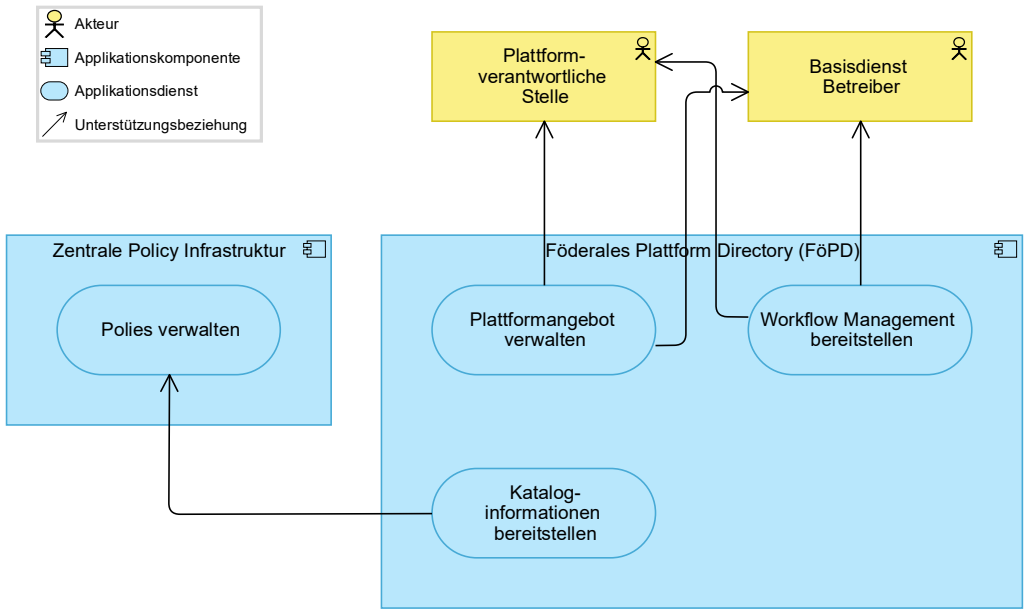


Abbildung 47: Use-Case-Nutzungsdiagramm für den Prozess „Plattformangebot festlegen“

Informationsfluss im Prozessablauf:

Abbildung 48: Informationsflussdiagramm für den Prozess „Plattformangebot festlegen“

Ablaufbeschreibung

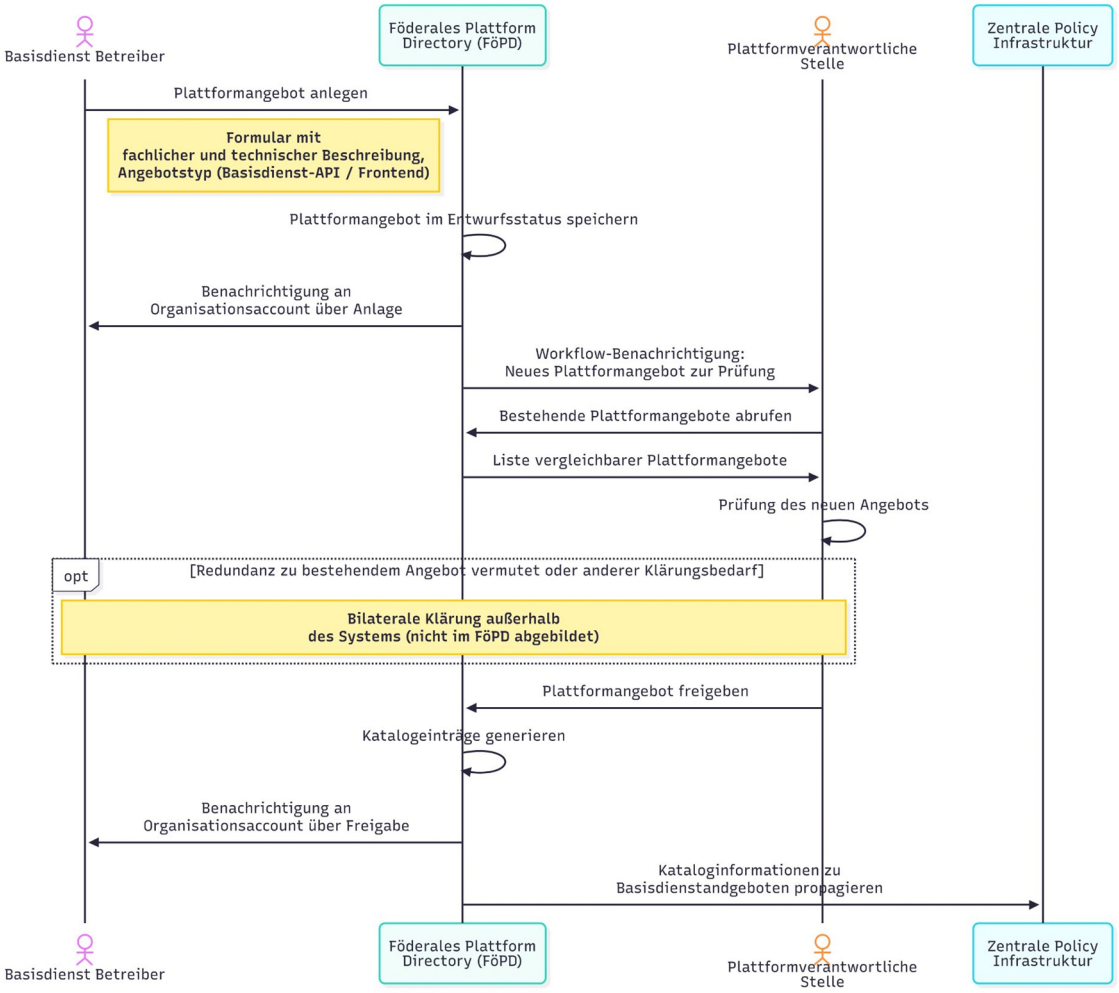


Abbildung 49: Sequenzdiagramm für den Prozess „Plattformangebot festlegen“

Der Prozess wird durch den Basisdienst Betreiber initiiert, der über das Föderale Plattform Directory ein neues Plattformangebot anlegt. Hierzu steht ein standardisiertes Formular bereit, in dem das Angebot fachlich und technisch beschrieben wird. Dabei wird unter anderem festgelegt, ob es sich um eine Basisdienst-API oder ein Frontend-Angebot handelt. Das FöPD speichert das Plattformangebot im Entwurfsstatus und benachrichtigt den Organisationsaccount des Basisdienst Betreibers über die Anlage.

Anschließend informiert das FöPD die Plattformverantwortliche Stelle per Workflow-Benachrichtigung über das neue Plattformangebot. Die Plattformverantwortliche Stelle ruft die bestehenden Plattformangebote ab und erhält vom FöPD eine Liste vergleichbarer Angebote. Auf dieser Grundlage prüft sie das neue Angebot. Wird eine Redundanz zu einem bestehenden Angebot vermutet oder besteht anderer Klärungsbedarf, erfolgt eine bilaterale Klärung zwischen der Plattformverantwortlichen Stelle und dem Basisdienst Betreiber außerhalb des Systems.

Nach abgeschlossener Prüfung gibt die Plattformverantwortliche Stelle das Plattformangebot im FöPD frei. Das FöPD generiert daraufhin die zugehörigen Katalogeinträge und benachrichtigt den Organisationsaccount des Basisdienst Betreibers über die Freigabe. Abschließend propagiert das FöPD die Kataloginformationen zu Basisdienstangeboten automatisch an die Zentrale Policy Infrastruktur.

4.4.2.11 Prozess „Berechtigungsmodell konfigurieren“

Der Prozess „Berechtigungsmodell konfigurieren“ gehört zur Prozessgruppe „Plattformangebote bereitstellen“ und unterstützt den Wertstrom „APIs von Basisdiensten bereitstellen“. Er umfasst die Definition, das Testen und die Freigabe der Policies und der zugehörigen technischen Konfiguration für ein konkretes Basisdienst-Angebot. Das freigegebene Berechtigungsmodell bildet die technische Grundlage für die Policy-basierte Zugriffskontrolle im späteren API-Betrieb.

Relevante Use Cases:

Tabelle 30: Use Cases des Prozesses „Berechtigungsmodell konfigurieren“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Workflow Management bereitstellen	Föderales Plattform Directory (FöPD)	Basisdienst Betreiber, Plattformverantwortliche Stelle	Unterstützung des Konfigurations- und Freigabeprozesses: Anlage und Verwaltung des Freigabevorgangs für ein Berechtigungsmodell, Weiterleitung der Freigabeanfrage an die plattformverantwortliche Stelle sowie Übermittlung des Freigabestatus an die Zentrale Policy Infrastruktur.
Policies verwalten	Zentrale Policy Infrastruktur	Basisdienst Betreiber	Anzeige bestehender Policies eines Basisdienst-Angebots und der verfügbaren berechtigungsrelevanten Attribute; Anlage, Bearbeitung und Aktivierung von Policies einschließlich automatisierter Prüfung von Qualitätsregeln während der Konfiguration.
Testdatensätze erstellen	Zentrale Policy Infrastruktur	Basisdienst Betreiber	Generierung von Testdatensätzen zum Erproben definierter Policies vor der produktiven Aktivierung.
Policies und berechtigungsrelevante Attribute verteilen	Zentrale Policy Infrastruktur	[Basisdienst] Dezentrale Policy Infrastruktur	Bereitstellung der konfigurierten Policies und der zugehörigen berechtigungsrelevanten Attribute an die dezentrale Policy Infrastruktur des Basisdienstes zu Testzwecken.

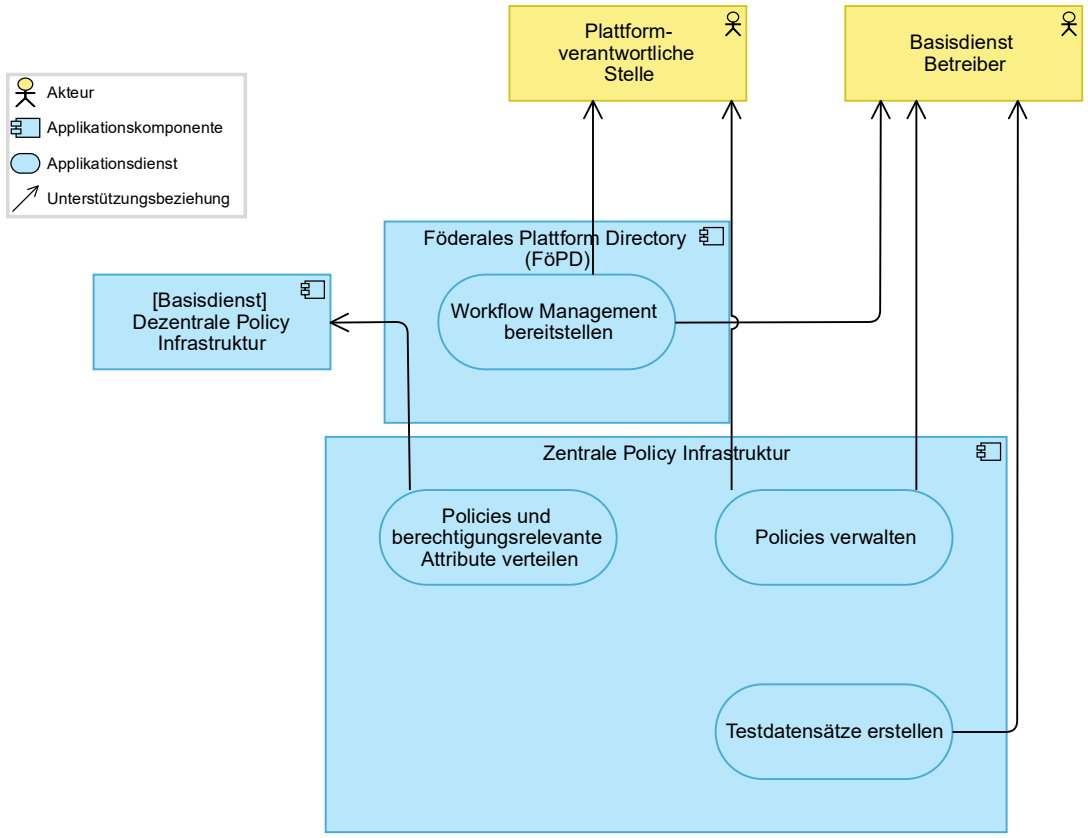


Abbildung 50: Use-Case-Nutzungsdiagramm für den Prozess „Berechtigungsmodell konfigurieren“

Informationsfluss im Prozessablauf:

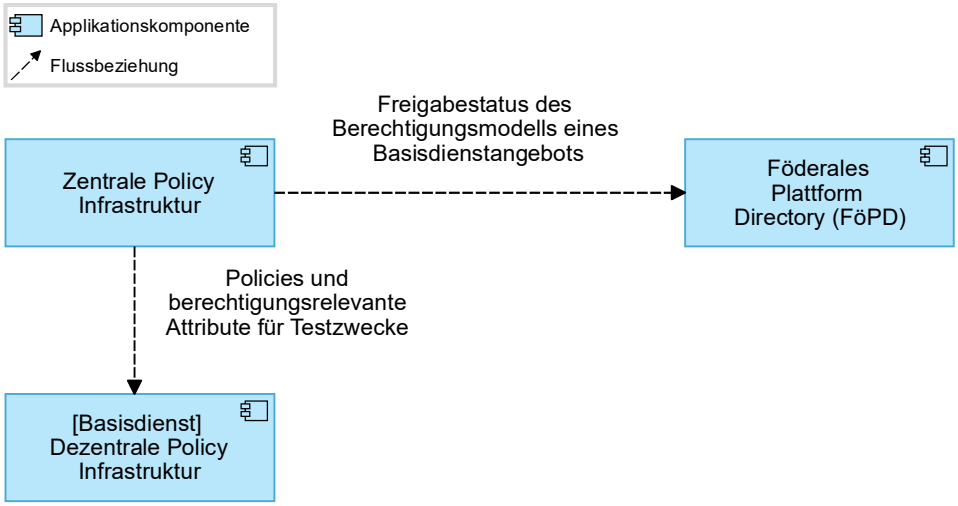


Abbildung 51: Informationsflussdiagramm für den Prozess „Berechtigungsmodell konfigurieren“

Ablaufbeschreibung

Der Prozess wird durch einen Basisdienst-Betreiber initiiert. Im Self-Service-Portal der Zentralen Policy Infrastruktur wählt er das Basisdienst-Angebot aus, für das das Berechtigungsmodell konfiguriert oder angepasst werden soll. Die Zentrale Policy Infrastruktur zeigt daraufhin die bereits bestehenden Policies des Berechtigungsmodells sowie die für das Angebot verfügbaren berechtigungsrelevanten Attribute an. Die Attributdefinitionen selbst stammen aus führenden Datenmanagementsystemen wie dem FIM-Datenfeldrepository, NOOTS DAMAS oder weiteren fachlichen Datenmanagementsystemen; die Definition neuer Attribute ist nicht Bestandteil dieses Prozesses und wird in einer künftigen Iteration der Architektur konzipiert.

Der Basisdienst-Betreiber definiert oder passt anschließend die Policies an, die das Berechtigungsmodell des Angebots ausmachen. Während der Konfiguration prüft die Zentrale Policy Infrastruktur automatisiert prüfbare Qualitätsregeln und meldet entsprechende Hinweise zurück.

Vor der produktiven Aktivierung können die Policies in einer Testumgebung des Basisdienstes erprobt werden. Hierfür generiert die Zentrale Policy Infrastruktur Testdatensätze und stellt die konfigurierten Policies sowie die zugehörigen berechtigungsrelevanten Attribute der dezentralen Policy Infrastruktur des Basisdienstes für Testzwecke bereit. Der Basisdienst-Betreiber konfiguriert und aktiviert die Policies in der Testumgebung und führt die Tests durch. Mit der Testumgebung ist hier eine Erprobungsumgebung für den Basisdienst selbst gemeint, nicht eine vom Basisdienst gegenüber API-Konsumenten bereitgestellte Testumgebung.

Nach erfolgreichem Test legt der Basisdienst-Betreiber die technischen Konfigurationen für die Verteilung der Policies fest und beantragt die Freigabe des Berechtigungsmodells für das Basisdienst-Angebot. Die Zentrale Policy Infrastruktur startet daraufhin einen Freigabevorgang bei der plattformverantwortlichen Stelle. Diese prüft das Berechtigungsmodell, beispielsweise auf Vollständigkeit, Konsistenz und weitere Qualitätsvorgaben, und gibt es bei positiver Entscheidung frei. Anschließend übermittelt die Zentrale Policy Infrastruktur den Freigabestatus an das Föderale Plattform Directory.

Perspektivisch ist vorgesehen, dass eine fachverbundverantwortliche Stelle eine Steuerungs- bzw. Abnahmefunktion wahrnehmen kann, sofern der Basisdienst Teil eines Fachverbunds ist. Die zugehörigen Detailregelungen werden in einer künftigen Iteration der Architektur konzipiert.

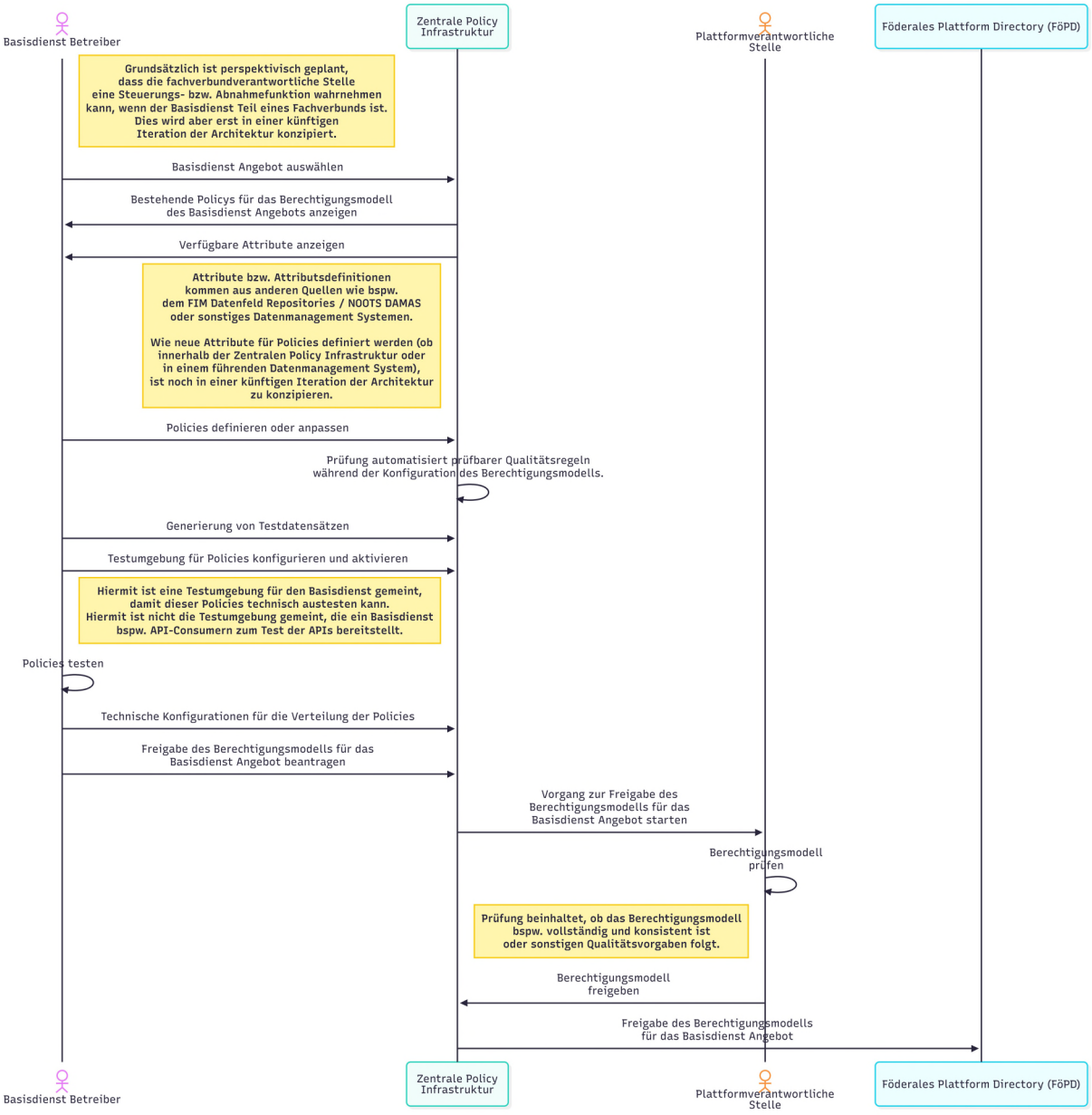


Abbildung 52: Sequenzdiagramm für den Prozess „Berechtigungsmodell konfigurieren“

4.4.2.12 Prozess „Kataloginformationen veröffentlichen“

Der Prozess „Kataloginformationen veröffentlichen“ gehört zur Prozessgruppe „Plattformangebote bereitstellen“ und unterstützt den Wertstrom „APIs von Basisdiensten bereitstellen“. Er umfasst die Beantragung und Freigabe der Veröffentlichung eines Basisdienst-Angebots, die Veröffentlichung des Katalogeintrags im Föderalen Plattform Directory sowie die anschließende Aktivierung des zuvor konfigurierten und freigegebenen Berechtigungsmodells für den

produktiven Einsatz. Mit der Veröffentlichung ist das Basisdienst-Angebot für berechtigte Organisationen sichtbar und nutzbar, womit der Bereitstellungsprozess abgeschlossen ist.

Relevante Use Cases:

Tabelle 31: Use Cases des Prozesses „Kataloginformationen veröffentlichen“

Use Case	Use Case Umsetzer	Use Case Nutzer	Kurzbeschreibung
Plattformangebot verwalten	Föderales Plattform Directory (FöPD)	Plattformverantwortliche Stelle	Verwaltung der Lebenszyklus-Stati eines Plattformangebots, hier insbesondere der Übergang in den veröffentlichten Zustand.
Workflow Management bereitstellen	Föderales Plattform Directory (FöPD)	Basisdienst Betreiber, Plattformverantwortliche Stelle	Unterstützung des Veröffentlichungsprozesses: Anlage und Verwaltung des Vorgangs zur Katalogfreigabe sowie Weiterleitung der Freigabeanfrage an die plattformverantwortliche Stelle.
Katalog-informationen bereitstellen	Föderales Plattform Directory (FöPD)	Basisdienst Betreiber	Veröffentlichung des Katalogeintrags eines Basisdienst-Angebots im Plattformkatalog nach erfolgter Freigabe.
Policies verwalten	Zentrale Policy Infrastruktur	Föderales Plattform Directory (FöPD)	Aktivierung des für ein Basisdienst-Angebot konfigurierten und freigegebenen Berechtigungsmodells für den produktiven Einsatz.
Policies und berechtigungsrelevante Attribute verteilen	Zentrale Policy Infrastruktur	[Basisdienst] Dezentrale Policy Infrastruktur	Bereitstellung der aktivierten Policies und der zugehörigen berechtigungsrelevanten Attribute an die dezentrale Policy Infrastruktur des Basisdienstes für den produktiven Einsatz.

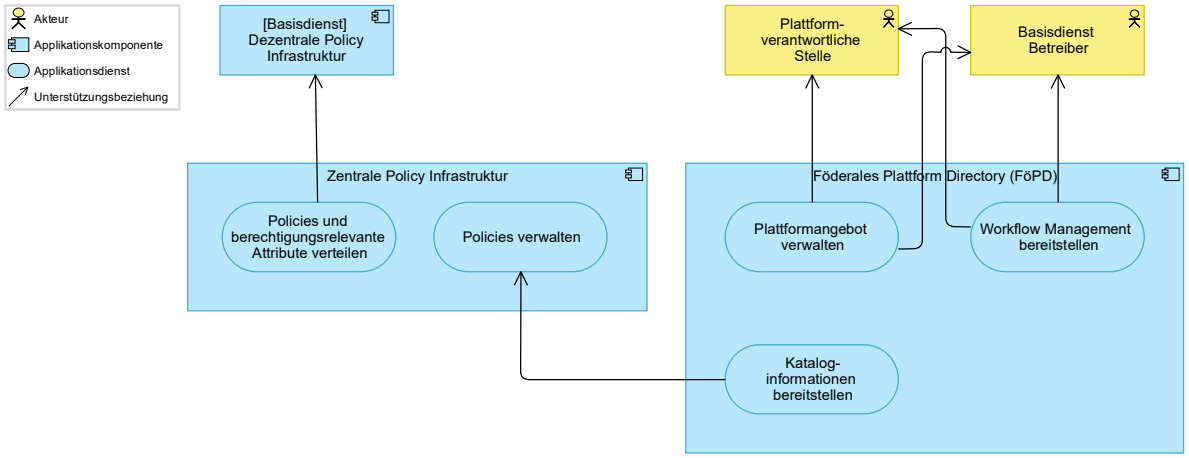


Abbildung 53: Use-Case-Nutzungsdiagramm für den Prozess „Kataloginformationen veröffentlichen"

Informationsfluss im Prozessablauf:

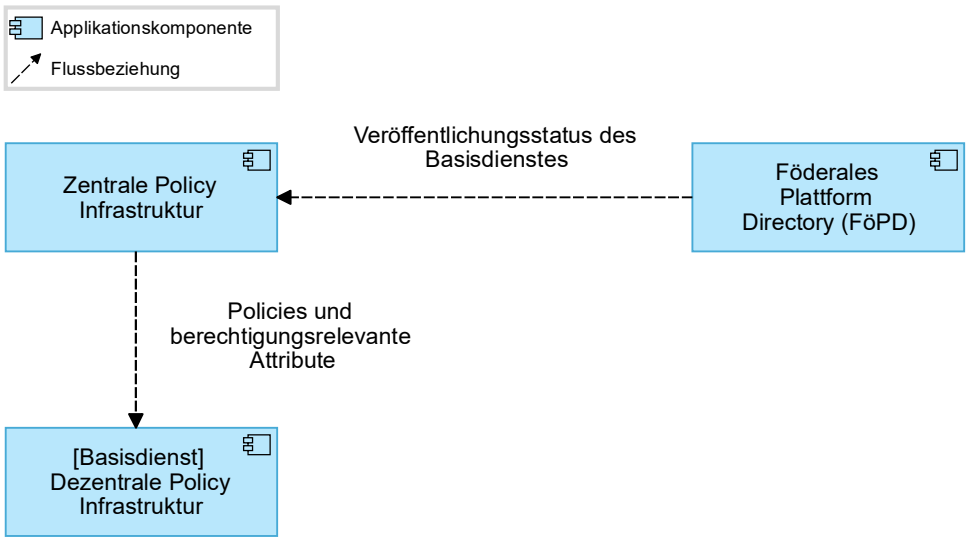


Abbildung 54: Informationsflussdiagramm für den Prozess „Kataloginformationen veröffentlichen"

Ablaufbeschreibung

Der Prozess wird durch einen Basisdienst-Betreiber initiiert. Über das Self-Service-Portal des Föderalen Plattform Directorys beantragt er die Veröffentlichung eines Basisdienst-Angebots im Plattformkatalog. Das FöPD prüft daraufhin die Vollständigkeit der Kataloginformationen und der ergänzenden Konfigurationen.

Anschließend startet das FöPD einen Freigabevorgang bei der plattformverantwortlichen Stelle. Diese prüft den beantragten Katalogeintrag und gibt die Veröffentlichung bei positiver Entscheidung frei. Das Berechtigungsmodell selbst ist im vorgelagerten Prozess „Berechtigungsmodell konfigurieren" bereits geprüft und freigegeben worden und ist hier nicht erneut Gegenstand der Prüfung.

Nach erfolgter Freigabe veröffentlicht das FöPD den Katalogeintrag und stößt über die Zentrale Policy Infrastruktur die Aktivierung des zugehörigen Berechtigungsmodells für den produktiven Einsatz an. Die Verteilung der Policies und der berechtigungsrelevanten Attribute an die dezentrale Policy Infrastruktur des Basisdienstes erfolgt anschließend durch die Zentrale Policy Infrastruktur und wird in diesem Konzept nicht weiter detailliert.

Mit der Veröffentlichung ist der Katalogeintrag für berechtigte Organisationen sichtbar und das Basisdienst-Angebot zur Nutzung bereit. Damit ist der Bereitstellungsprozess eines Plattformangebots abgeschlossen.

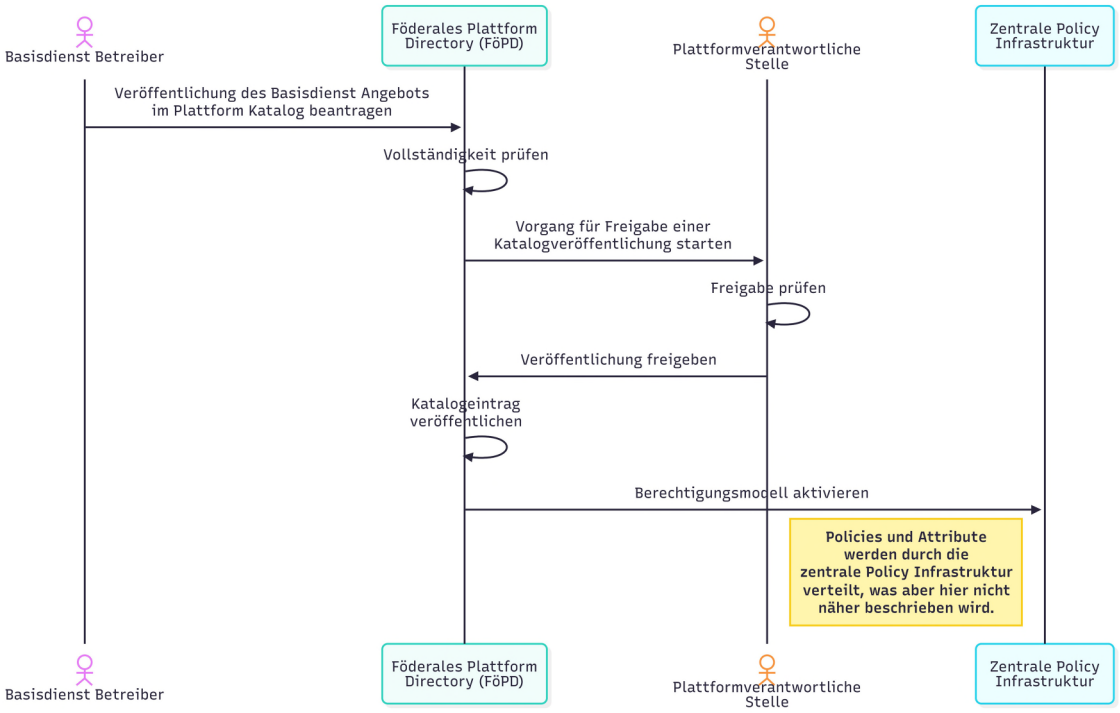


Abbildung 55: Sequenzdiagramm für den Prozess „Kataloginformationen veröffentlichen"

4.4.2.13 Prozess „Externe Organisationsattribute synchronisieren“

Der Prozess „Externe Organisationsattribute synchronisieren“ betrifft die Übernahme von Organisationsattributen aus externen Quellsystemen in das Föderale Plattform Directory. In der vorliegenden Version des Konzepts konnte dieser Prozess noch nicht spezifiziert werden; eine Ausarbeitung erfolgt im Rahmen künftiger Iterationen der Architektur.

4.4.3 Ergänzende Hinweise zu den Supportprozessen

Die nachfolgende Übersicht enthält die Supportprozesse der Zielarchitektur in alphabetischer Reihenfolge. Eine vertiefte Betrachtung dieser Prozesse ist nicht Gegenstand dieses Konzepts.

Tabelle 32: Supportprozesse der Zielarchitektur

Prozess	Kurzbeschreibung
Basisdienst-Angebot verwalten	Anlage, Änderung und Außerbetriebnahme einzelner Basisdienst-Angebote im FöPD.
Externe Nutzer berechtigen	Vergabe und Entzug von FöPD-Berechtigungen für einzelne FöPD-Nutzer.
Externe Organisationen berechtigen	Vergabe und Entzug von FöPD-Berechtigungen für externe Organisationen.
FöPD Accounts sperren	Sperrung einzelner FöPD-Accounts.
FöPD Nutzeraccounts verwalten	Anlage, Änderung und Deaktivierung von FöPD-Nutzeraccounts.
FöPD Zertifikat rotieren	Planmäßiger Austausch der vom FöPD verwendeten Signaturzertifikate.
FöPD Zertifikat sperren	Sperrung von FöPD-Signaturzertifikaten im Sperrfall.
Plattformangebot verwalten	Anlage, Änderung und Außerbetriebnahme von Plattformangeboten im FöPD.
Software sperren	Sperrung registrierter Software-Anwendungen im FöPD.
Software verwalten	Anlage, Änderung und Deaktivierung von Softwareeinträgen im FöPD.

4.5 Übersicht der umzusetzenden Use Cases

Die nachfolgende Tabelle fasst alle in den Prozesskapiteln identifizierten Use Cases zusammen. Für jeden Use Case werden das umsetzende System, die unterstützten Prozesse, die zugeordnete Fähigkeit aus dem Fähigkeitenmodell sowie die verwendeten Standardprotokolle und Schnittstellen aufgeführt. Use Cases, die in mehreren Prozessen auftreten, werden konsolidiert dargestellt.

Tabelle 33: Umzusetzende Use Cases

Use Case	Kurzbeschreibung	Umsetzendes System	Unterstützter Prozess	Unterstützte Fähigkeit	Verwendete Standardprotokolle und Schnittstellen
Organisation registrieren	Entgegennahme der Registrierungsanfrage, Koordination des Registrierungsprozesses	FöPD	4.4.2.1	Identitätsmanagement	
FöPD Nutzer authentifizieren	Authentifizierung eines beim FöPD registrierten Nutzers	FöPD Identity Provider	4.4.2.1, 4.4.2.6	Identitätsmanagement	OIDC
Identitäten überprüfen	Überprüfung und Zusammenführung der Identitätsnachweise	FöPD Identity Provider	4.4.2.1	Identitätsmanagement	
Accounts verwalten	Anlage/Aktualisierung von Organisations- und Nutzeraccounts	FöPD Identity Provider	4.4.2.1, 4.4.2.2	Identitätsmanagement	
Natürliche Personen authentifizieren	Authentifizierung natürlicher Personen via BundID	BundID	4.4.2.1, 4.4.2.7	Identitätsmanagement	SAML
Juristische Personen authentifizieren	Authentifizierung juristischer Personen via MUK	Mein Unternehmenskonto (MUK)	4.4.2.1, 4.4.2.7	Identitätsmanagement	SAML
Natürliche oder juristische Person authentifizieren	Authentifizierung über externe föderale Identity Provider	Externe Identity Provider	4.4.2.7	Identitätsmanagement	OIDC/SAML

Organisationseigenschaften verwalten	Anzeige und Beantragung von Organisationseigenschaften	FöPD	4.4.2.3	Identitätsmanagement	
Workflow Management bereitstellen	Anlage/Verwaltung von Vorgängen, Benachrichtigung Beteiligter	FöPD	4.4.2.2, 4.4.2.3, 4.4.2.10, 4.4.2.11, 4.4.2.12	Workflow Management	
Policy relevante Informationen verwalten	Bereitstellung/Verwaltung Policy-relevanter Informationen	Zentrale Policy Infrastruktur	4.4.2.2	Berechtigungsmanagement	
Policy prüfen (zentral)	Auswertung einer Policy-Abfrage und Rückgabe einer Policy-Entscheidung	Zentrale Policy Infrastruktur	4.4.2.2	Berechtigungsmanagement	AuthZEN
Policy prüfen (dezentral)	Auswertung der Berechtigungsanfrage und Rückgabe einer Policy-Entscheidung	[Basisdienst] Dezentrale Policy Infrastruktur	4.4.2.5, 4.4.2.6, 4.4.2.8, 4.4.2.9	Berechtigungsmanagement	AuthZEN
Identitätsdaten bereitstellen	Bereitstellung aktueller Organisationsattribute für Policy Infrastruktur	FöPD Identity Provider	4.4.2.2, 4.4.2.3	Identitätsmanagement	SCIM
Identitätsdaten aktualisieren	Speicherung/Aktualisierung der Organisationsattribute im IdP	FöPD Identity Provider	4.4.2.3	Identitätsmanagement	SCIM
Attribute Store verwalten	Verwaltung der für Berechtigungsregeln relevanten Attribute	Zentrale Policy Infrastruktur	4.4.2.3, 4.4.2.4	Berechtigungsmanagement	
Software verwalten	Anlage einer Software im FöPD mit APIs, Berechtigungen, Metadaten	FöPD	4.4.2.4	API-Management	

M2M Client autorisieren	Optionale M2M-Autorisierung der Software durch Basisdienst Nutzer	FöPD	4.4.2.4	Autorisierungsmanagement	
Software Informationen bereitstellen	Bereitstellung Software-Informationen an Zentrale Policy Infrastruktur	FöPD	4.4.2.4	Berechtigungsmanagement	
Software Statement Assertion ausstellen	Generierung und Signierung der SSA	FöPD	4.4.2.4	Vertrauensinfrastruktur- und Zertifikatsmanagement	RFC 7591
API-Clients registrieren und verwalten	Entgegennahme Client Registration, Prüfung SSA, Clientanlage	[Basisdienst] Authorization Server	4.4.2.5	API-Management	RFC 7591 (DCR), FAPI 2.0
Nutzer weiterleiten	Anzeige verfügbarer Frontend-Angebote, Weiterleitung zum Basisdienst	FöPD	4.4.2.6	Frontendbereitstellung	
Frontend Funktionalitäten bereitstellen	Entgegennahme Zugriffsanfrage, OIDC Flow, Attributbereitstellung	Basisdienst	4.4.2.6		OIDC
Nutzerautorisierung erfassen	Erfassung der expliziten Nutzereinwilligung für API-Zugriff	Authorization Server für Nutzerzustimmung	4.4.2.7	Autorisierungsmanagement	OAuth 2.0, FAPI 2.0, PKCE
API-unspezifischen Access Token ausgeben	Ausstellung eines DPoP-gebundenen Opaque Access Tokens nach Nutzerautorisierung	Authorization Server für Nutzerzustimmung	4.4.2.7	Autorisierungsmanagement	OAuth 2.0, FAPI 2.0, DPoP
API Access Token für Client Credentials ausstellen	Ausstellung eines DPoP-gebundenen Opaque Access Tokens via Client Credentials	[Basisdienst] Authorization Server	4.4.2.8	Autorisierungsmanagement	OAuth 2.0, FAPI 2.0, DPoP, `private_key_jwt`
API Access Token für Token Exchange ausstellen	Ausstellung eines API-spezifischen Tokens durch Token Exchange	[Basisdienst] Authorization Server	4.4.2.8	Autorisierungsmanagement	OAuth 2.0 Token Exchange (RFC 8693), FAPI 2.0, DPoP

Opaque Access Token auflösen (Nutzerzustimmung)	Auflösung des Tokens via Token Introspection, Rückgabe JWS Access Token	Authorization Server für Nutzerzustimmung	4.4.2.8	Autorisierungsmanagement	Token Introspection (RFC 7662)
Opaque Access Token auflösen (Basisdienst)	Entgegennahme Opaque Token via Introspection, Generierung JWS	[Basisdienst] Authorization Server	4.4.2.9	Autorisierungsmanagement	Token Introspection (RFC 7662)
API-Anfragen vorprüfen	Entgegennahme API-Aufruf, Token-Auflösung, DPoP-Prüfung, Berechtigungsprüfung	[Basisdienst] API-Gateway	4.4.2.9	API-Management	DPoP, Token Introspection
JWS Token abrufen	Auflösung Opaque Token beim AS, Caching für künftige Requests	[Basisdienst] API-Gateway	4.4.2.9	API-Management	Token Introspection (RFC 7662)
API bereitstellen	Verarbeitung des API-Aufrufs mit JWS Access Token	Basisdienst	4.4.2.9		
Plattformangebot verwalten	Anlage, Bearbeitung und Freigabe von Plattformangeboten	FöPD	4.4.2.10, 4.4.2.12	API-Management	
Kataloginformationen bereitstellen	Generierung und Freigabe Katalogeinträge, Propagierung an Zentrale Policy Infrastruktur	FöPD	4.4.2.2, 4.4.2.10, 4.4.2.12	API-Management	
Policies verwalten	Anzeige, Anlage, Bearbeitung und Aktivierung von Policies eines Basisdienst-Angebots einschließlich automatisierter Prüfung von Qualitätsregeln	Zentrale Policy Infrastruktur	4.4.2.10, 4.4.2.11, 4.4.2.12	Berechtigungsmanagement	
Testdatensätze erstellen	Generierung von Testdatensätzen zum Erproben definierter Policies vor der produktiven Aktivierung	Zentrale Policy Infrastruktur	4.4.2.11	Berechtigungsmanagement	

Policies und berechtigungsrelevante Attribute verteilen	Bereitstellung der konfigurierten Policies und der zugehörigen berechtigungsrelevanten Attribute an die dezentrale Policy Infrastruktur eines Basisdienstes	Zentrale Policy Infrastruktur	4.4.2.11, 4.4.2.12	Berechtigungsmanagement	

5 Querschnittliche Themen

5.1 Berechtigungskonzept

5.1.1 Ausgangslage, Zielsetzung

Die Ausgangslage für das vorliegende Berechtigungskonzept ist durch die erforderliche die Zugriffssteuerung auf föderale API-Schnittstellen gegeben, das die Berechtigung von API-Clients darauf prüft, Aktionen auf föderalen APIs auszuführen.

5.1.1.1 Zielsetzung

Die Steuerung des Zugriffs auf APIs in einer heterogenen, behördenübergreifenden Infrastruktur erfordert ein flexibles Berechtigungskonzept, das den vielfältigen Anforderungen der Praxis gerecht wird.

Die Breite der adressierten Nutzerschaft – sowohl auf Seiten der API-Anbieter als auch der API-Nutzer – geht mit einer Vielzahl möglicher Anwendungsszenarien einher, die sich weder vollständig antizipieren noch auf ein einheitliches Fachmodell reduzieren lassen. Hinzu kommen dezentrale Organisationsstrukturen, in denen Betreiber von APIs und Basisdiensten den Zugriff eigenverantwortlich für spezifizierte Nutzergruppen freigeben müssen – ohne dabei jeden einzelnen Nutzer explizit und individuell berechtigen zu können. Das Konzept muss es daher ermöglichen, den Kreis potenzieller Nutzer anhand beliebiger Kriterien zu definieren und diesen je nach Anforderung eng oder weit einzugrenzen.

Grundlegend für das Konzept ist die Unterscheidung zwischen zwei Ebenen der Zugriffssteuerung:

- Auf der **API-Ebene** wird in einem groben Zuschnitt („coarse-grained") geregelt, welcher API-Client auf welches API zugreifen darf.
- Auf der **Anwendungsebene** hingegen wird feingranular („fine-grained") bestimmt, welche konkreten Berechtigungen ein API-Client innerhalb der jeweiligen Anwendung besitzt.

Das vorliegende Konzept beschreibt ein regelbasiertes, attributgetriebenes Berechtigungsmodell zur Zugriffssteuerung auf APIs. Es umfasst zwei aufeinander aufbauende Ebenen:

- die grobgranulare Zugangssteuerung (coarse-grained authorization) – ob eine Client-Software überhaupt Zugang zu einer API erhält

- sowie eine scope-basierte Berechtigungsebene, die dem Resource Server (API) ausreichend Kontext für feingranulare Entscheidungen auf Anwendungsebene liefert.
- Vollständig feingranulare Zugriffssteuerung unterhalb der Scope-Ebene (z. B. auf Ressourcenpfad- oder Feldebene) verbleibt in der Verantwortung des jeweiligen API-Betreibers.

Das vorliegende Konzept konzentriert sich auf die API-Ebene. Im Mittelpunkt stehen dabei die Zugriffssteuerung von Clients auf APIs, die Definition eines Modells zur Steuerung und Vergabe von Berechtigungen durch berechtigte Stellen – insbesondere Betreiber von Basisdiensten und APIs (Anm. „Berechtigte Stelle“) – sowie die Beschreibung der Schnittstellen, über die die Berechtigungsprüfung von API-Clients technisch realisiert wird.

Für die Modellierung des Berechtigungssystems sind folgende Leitlinien prägend:

- Dezentralität: API-Betreiber verwalten ihre Berechtigungsregeln fachlich eigenverantwortlich.
- Flexibilität: Das Modell ist fachdomänenunabhängig und unterstützt vielfältige Nutzungsszenarien.
- Nachvollziehbarkeit: Berechtigungsentscheidungen sind transparent und auf leicht verständliche Regeln zurückführbar.
- Zentrales Vertrauen: Das föderale Platform-Directory stellt für die Berechtigungsverwaltung relevante Informationen über Software Instanzen (für API-Clients und APIs von Basisdiensten) in Form von Attributinformationen bereit.

5.1.1.2 Abgrenzung Berechtigungsprüfung / Architektur

Das Berechtigungskonzept basiert auf einem regelbasierten, attributgetriebenen Modell zur Zugriffssteuerung auf APIs. Aussagen über feingranulare Zugriffssteuerung innerhalb eines API-Dienstes (etwa auf Ressourcen- oder Feldebene) sind nicht Gegenstand dieses Konzepts; diese verbleiben in der Verantwortung des jeweiligen API-Betreibers.

Die Granularität, in der eine Berechtigungsanfrage geprüft wird, hängt von der Komponente ab, die den Client-Request verarbeitet. Es werden zwei Stufen unterschieden:

- **Grobgranulare Prüfung (Coarse-grained)** findet am **Authorization Server** des Basisdienstes statt. Sie wird einmalig ausgelöst, wenn ein API-Client ein Access Token für eine API des Basisdienstes anfordert. Gegenstand der Prüfung ist die Frage, ob der API-Client berechtigt ist, mit einer bestimmten API des Basisdienstes zu kommunizieren. Zusätzlich

dazu prüft auch das Ziel-API in jedem Moment des Zugriffs (bzw. dessen vorgelagertes API-Gateway), ob die weitere Kommunikation überhaupt zulässig ist.

- **Feingranulare Prüfung (Fine-grained)** erfolgt innerhalb der Anwendung bzw. des API-Dienstes selbst – bei jeder eingehenden, gültig authentifizierten Anfrage eines Clients. Sie beantwortet die Frage, welche Aktionen der Client auf welchen Ressourcen innerhalb der Anwendung ausführen darf. Grundlage dieser Prüfung sind das Access Token des Clients, die angeforderte Aktion, die betroffene Ressource sowie relevante Kontextdaten.

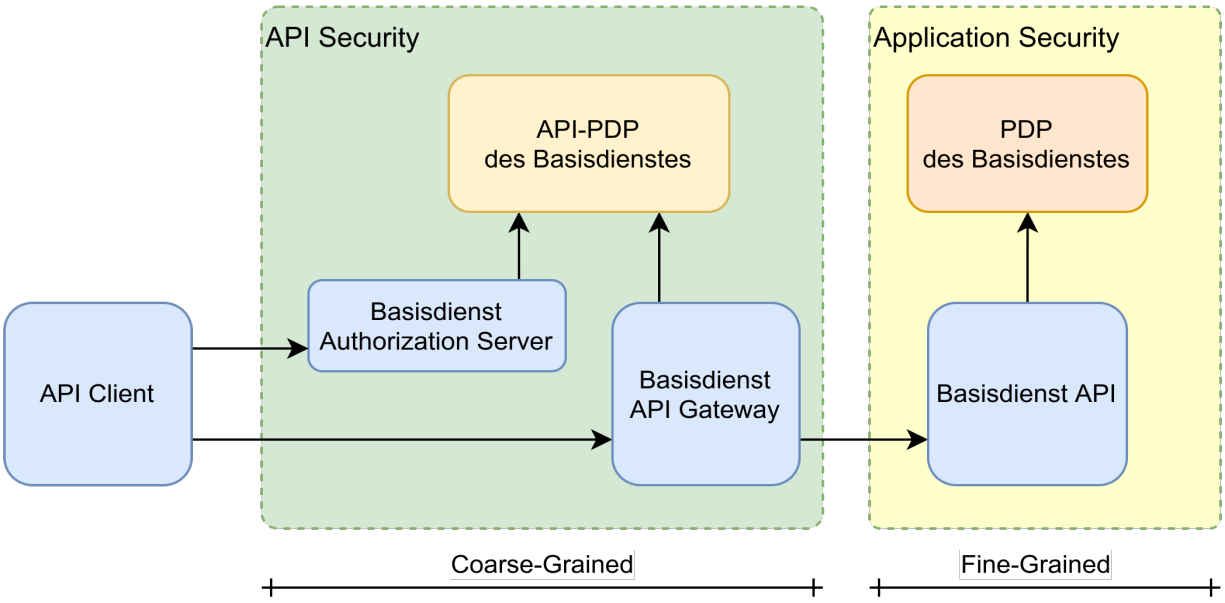


Abbildung 56: Stufen des Berechtigungskonzepts

Das vorliegende Berechtigungskonzept betrachtet ausschließlich die grobgranulare Ebene, also die Steuerung des Zugriffs von API-Consumer-Clients auf APIs von Basisdiensten.

Dies hat folgende, wesentliche Gründe:

- Feingranulare Berechtigungssteuerung setzt ein tiefes Verständnis der fachlichen Zusammenhänge der jeweils zu schützenden Anwendung voraus, das auf übergreifender Konzeptebene nicht vorausgesetzt werden kann.

- Es sollen technische Vorgaben vermieden werden, die in die interne Verarbeitung und Berechtigungslogik einzelner Anwendungen und APIs eingreifen und dort bestehende Steuerungsmechanismen beeinträchtigen könnten.

Abgesehen von dieser Abgrenzung ist das Berechtigungssystem bewusst offen und generisch ausgelegt: Es kann an verschiedenen Stellen um zusätzliche Anwendungsfälle erweitert und ergänzt werden, ohne dass die Grundarchitektur angepasst werden muss.

Die Systemarchitektur ist weitgehend modular aufgebaut und orientiert sich an etablierten Best Practices und Architekturmustern:

- Für die wesentlichen Sicherheitskomponenten werden etablierte Architekturmuster (PDP, PEP, PIP und PAP) verwendet, um Berechtigungen zu definieren und auf API-Ebene durchzusetzen.
- Die wesentlichen Kernfunktionen werden durch dedizierte und voneinander entkoppelte Komponenten realisiert, so dass die Auswirkungen eines „assume breach" Szenarios minimiert werden
- Für die Identitätsschicht wird auf OAuth 2.0 und OpenID Connect (OIDC) gesetzt; komplementär dazu wird das OpenID AuthZEN-Protokoll für die Prozessschnittstellen in der Berechtigungsschicht verwendet.

Dies gewährleistet weitgehende Kompatibilität der relevanten Komponenten, vermeidet Abhängigkeiten von Herstellern oder Implementierungen und stellt technologische Neutralität sicher.

Entwickler und Betreiber von Plattformprodukten werden dazu ermuntert, die hier beschriebenen Architekturmuster und Plattformschnittstellen der Berechtigungsschicht als Grundlage für eigene, feingranulare Berechtigungskonzepte auf Anwendungsebene zu adaptieren und in die Systemumgebung zu integrieren.

5.1.2 Überblick Systemarchitektur

Die Berechtigungsarchitektur der Plattform besteht aus mehreren zusammenwirkenden Komponenten, die gemeinsam die regelbasierte Zugriffssteuerung auf API-Ebene realisieren.

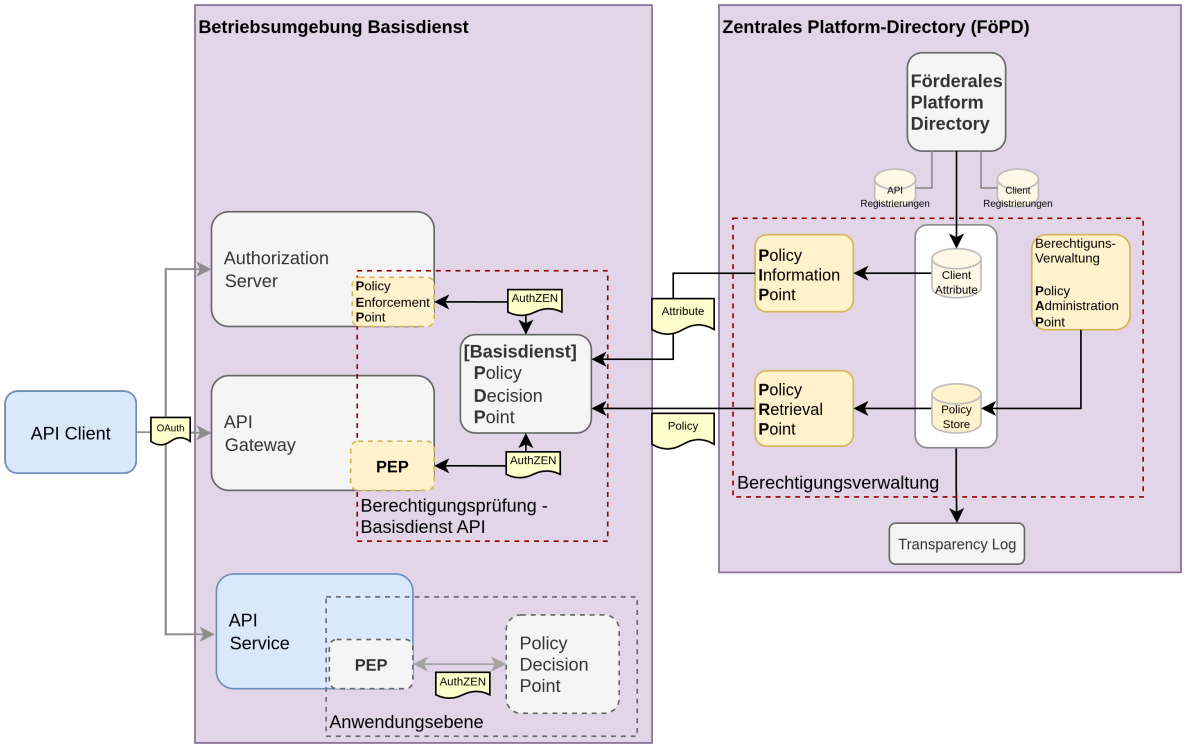


Abbildung 57: Berechtigungsarchitektur

Systemkomponenten und Interaktion

Das zentrale **Platform Directory (FöPD)** bildet den Vertrauensanker der gesamten Architektur. Es ist die autoritative Quelle für die Registrierung aller APIs und Clients über Software Statements, für den zentralen Attributkatalog sowie für die Verwaltung von Berechtigungsregeln. Der **Policy Administration Point (PAP)** ist die administrative Schnittstelle der Berechtigungsschicht, über die API-Verantwortliche und Plattformadministratoren Berechtigungsregeln verwalten.

Die im PAP verwalteten Policies werden durch den Policy Retrieval Point (RPR) aufbereitet und gezielt an die Betreiber-PDPs ausgeliefert (Distribution).

Der **Policy Information Point (PIP)** ist die administrative Komponente in der Berechtigungsschicht, über die Attribute über berechtigungsfähige Softwarekomponenten (Clients, Software Statements) verwaltet werden. Der PIP bezieht diese Daten in Form von Attributen und Metadaten vom Platform Directory und bereitet sie so auf, dass sie an die dezentralen Policy Decision Points ausgeliefert werden können. Der PIP versorgt so die Betreiber-PDPs mit aktuellen Attributdaten aus dem Platform Directory.

Änderungen an Policies oder Attributdaten werden jeweils unabhängig voneinander an die zuständigen PDP-Instanzen ausgeliefert – jeder PDP erhält ausschließlich die Policies und Attribut-Informationen, die seinem Zuständigkeitsbereich entsprechen.

Auf Seiten der Basisdienste ist der **Authorization Server (AS)** die zentrale Vertrauenskomponente: Er stellt API-Clients Access Tokens aus und delegiert die Berechtigungsentscheidung an den **Policy Decision Point (PDP)**. Der PDP wertet den für das jeweilige API gültigen Regelbestand aus und gibt eine Entscheidung zurück. Der **Policy Enforcement Point (PEP)** – realisiert am AS oder einem vorgelagerten API-Gateway – setzt die Entscheidung des PDP durch und gewährt oder verweigert den Zugriff.

Die Systemgrenzen verlaufen zwischen dem zentralen Platform Directory und den dezentralen Basisdiensten: Das Platform Directory verantwortet die Registrierung, Attributverwaltung und Policy-Administration; die Basisdienste verantworten die operative Berechtigungsprüfung und Token-Ausstellung im eigenen Zuständigkeitsbereich.

Die funktionale Trennung zwischen PAP (Administration), PIP und PRP (Distribution) ermöglicht eine zusätzliche Härtung der Systemarchitektur und reduziert die Angriffsfläche durch das Prinzip der minimalen Exposition: Der PAP und das FöPD sind ausschließlich für autorisierte Administratoren über gesicherte, stark authentifizierte Kanäle erreichbar und vom Laufzeitpfad der Policy-Auswertung vollständig entkoppelt. Kompromittierungen im Bereich der Policy-Auslieferung oder -Auswertung können so nicht direkt auf die administrative Verwaltungsebene durchschlagen.

Logisches Berechtigungsmodell (ABAC/PBAC)

Das logische Berechtigungsmodell kombiniert attributbasierte Zugriffssteuerung (ABAC) mit einem Policy-basierten Ansatz (PBAC). Anstatt einzelne Clients explizit zu berechtigen, können API-Betreiber Policies definieren, die den Zugriff anhand von Attributen der beteiligten Subjekte (Client, Benutzer) und Ressourcen (API) – steuern. Die Attribute entstammen den im Platform Directory registrierten Software Statements sowie ergänzenden Laufzeitinformationen, die der AS zur Entscheidungszeit beisteuert. Dieses Modell ermöglicht es, den Kreis berechtigter Nutzer flexibel und ohne individuelle Einzelberechtigungen zu definieren – und damit sowohl enge als auch weite Zugriffsprofile abzubilden, ohne jeden Client namentlich kennen zu müssen. Durch die Trennung zwischen Policies und Attributen wird eine flexible, dezentral verwaltbare Berechtigungssteuerung ermöglicht.

5.1.3 Berechtigungsmodell

5.1.3.1 Technische Begriffe im Berechtigungsmodell

Das Berechtigungskonzept verwendet eine Reihe domänenspezifischer Begriffe, deren Bedeutung im vorliegenden Abschnitt nachfolgend beschrieben wird.

Subject / Subjekt:

Ein Subject repräsentiert einen technischen Akteur, der eine Aktion im Berechtigungsmodell ausführen will. Es wird mindestens durch einen Typ und eine eindeutige ID definiert, und kann optional durch zusätzliche Attribute beschrieben werden.

Beispiel:

```
{
  "type": "software-statement",
  "id": "urn:platform-directory:ss:a1b2c3d4-e5f6-…",
  "properties": {
    "client.type": "onlinedienst",
    "acting_for": {
       "org_type": "behörde",
         "org_funktionskennzeichen": "fkz-001"
    }
  }
}
```

Im Kontext des vorliegenden Konzepts kann ein Subject sowohl konkrete Software-Instanzen repräsentieren, welche zur Laufzeit durch API-Clients repräsentiert werden, als auch z. B. individuelle Benutzer.

Beispiele für Subjekte:

- eine Software Instanz, die von einem zugehörigen API-Client repräsentiert wird.
- ein API-Client der nicht-interaktiv eine Aktion an einem API ausführen möchte
- ein API-Client der durch einen Endanwender (interaktiv) genutzt wird

Resource / Ressource:

Eine Resource ist das Ziel einer Zugriffsanforderung, beziehungsweise einer Aktion. Es handelt sich um ein Objekt, das ähnlich wie eine Subject-Entität aufgebaut ist und mindestens durch

einen **Typ** und eine eindeutige **ID** definiert wird. Optional kann es durch zusätzliche Attribute beschrieben werden.

Beispiel:

```
{
  "type": "api",
  "id": "https://example-api/endpoint-url",
  "properties": {
    ...
  }
}
```

Je nach dem, an welcher Stelle/durch welche Komponente im Prüfungsprozess die Berechtigungsprüfung stattfindet, werden unterschiedliche Entitäten durch eine Ressource repräsentiert und von der Zugriffssteuerung geschützt:

- **Coarse-Grained / API-Ebene:**

  Die Ressource repräsentiert in der Regel eine Software Instanz respektive API-Endpunkt, auf welche ein Client zugreifen möchte.

- **Fine-Grained / Anwendungsebene:**

  Die Ressource repräsentiert in der Regel ein beliebiges Objekt innerhalb der Anwendung, z. B. ein Dokument, eine Verwaltungsakte, oder ein beliebiges anderes Objekt.

Action / Aktion:

Eine Aktion ist die Art des Zugriffs, beziehungsweise die Funktion, die das ein Subject auf einer Resource ausführen möchte. Eine Aktion wird mindestens durch einen eindeutigen Namenschlüssel identifiziert.

Optionale Argumente der Funktion/Aktion werden als deren Eigenschaften ausgedrückt.

Beispiel:

```
{
  "name": "token_exchange",
  "properties": {
      "requested_scopes": ["lesen","schreiben"]
  }
}
```

Beispiele für Aktionen:

- im HTTP-Protokoll die Aktionen: GET, POST, PUT, DELETE, etc.
- ein Funktionsaufruf in einem gRPC API, z. B. „`print_document`“
- Eine Aktion innerhalb der Anwendung, z. B. „Dokument drucken“

5.1.3.2 Berechtigungsmodell

Im Föderalen Platform-Directory (FöPD) werden Berechtigungen für API-Clients in Form eines strukturierten Regelwerks – einer **Policy** – abgebildet. Das Berechtigungsmodell basiert auf vier zentralen Elementen:

Die grundlegenden Elemente der Berechtigungslogik sind:

- **Eine Policy** fasst Bedingungen und Zugriffsregeln (Conditions) für genau eine (API-)-Ressource, oder eine Gruppe von API-Ressourcen, zusammen und legt das Gesamtverhalten der Zugriffssteuerung fest.
- **Eine API-Ressource** verknüpft die Policy mit dem konkreten API-Endpunkt, für den sie gilt.
- **Conditions** sind atomare Prüfeinheiten innerhalb einer Policy: Sie bewerten einzelne Bedingungen – etwa Client-Attribute – und liefern einen booleschen Wahrheitswert (TRUE / FALSE).
- **Scopes** definieren die Menge möglicher Berechtigungen, die einem Client für die Nutzung einer API gewährt werden. Sie überbrücken die Lücke zwischen der grobgranularen Zugangsentscheidung und der feingranularen Anwendungslogik.

Die Policy bildet das zentrale Steuerungselement des Modells und wird im folgenden Abschnitt detailliert beschrieben.

Policy

Eine Policy ist das Kernelement des Berechtigungsmodells. Als geordnetes Containerobjekt fasst sie Regeln und Bedingungen für genau eine API zusammen und legt – sofern diese erfüllt sind – die Zugriffsentscheidung verbindlich fest.

Eine Policy wird als Datenstruktur repräsentiert, die folgende Elemente umfasst:

1 **API Resource:** Gibt die durch die Policy zu schützenden Ressourcen (Anwendung resp. API) an. Hierbei kann es sich um eine einzelne API-Ressource, oder eine Gruppe mehrere APIs handeln. Eine Policy kann immer einer zu schützenden API-Ressource(n-Gruppe) zugeordnet sein. Dennoch können es mehrere Policies für die-selbe API-Ressource(n-Gruppe) geben.

2 **Policy Effect (Auswirkung):** Gibt das gewünschte Ergebnis der Richtlinie an, wenn diese auf je jeweilige Zugriffsanfrage angewendet wird und die in der Richtlinie festgelegten Bedingungen zutreffen.

3 **Policy Conditions:** Eine Liste von Bedingungen. Diese Bedingungen müssen erfüllt sein, damit der Effect der Policy auf die Autorisierungsanforderung angewendet wird. Bedingungen werden als „wenn"-Regeln ausgedrückt. Innerhalb einer Policy müssen immer alle Bedingungen erfüllt sein.

4 **Policy Scopes:** Gibt an, welche Scopes diese Policy freigibt, wen sie zutrifft.

5 **Policy Exceptions:** Eine Liste von Bedingungen, mit denen eine zutreffende Policy außer Kraft gesetzt werden, wenn eine der Bedingungen zutrifft.

Policy Exceptions sind ausschließlich bei DENY-Policies semantisch gültig. Bei PERMIT-Policies wird es ignoriert bzw. darf nicht befüllt werden.

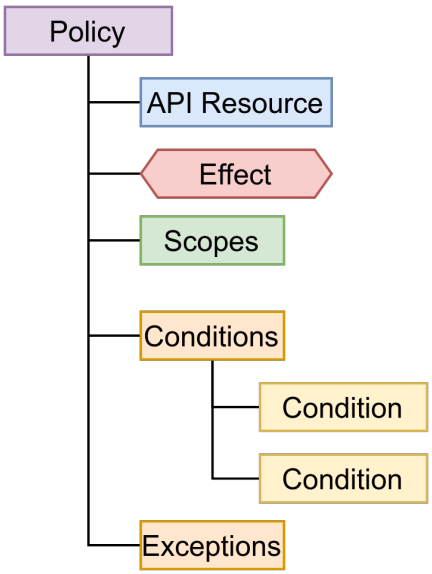


Abbildung 58: Datenstruktur einer Policy

Das Feld **scopes** ist ausschließlich bei PERMIT-Policies semantisch gültig. Jeder hier referenzierte Scope muss bei der Registrierung der API im FöPD deklariert worden sein – eine Policy kann ausschließlich Scopes gewähren, die die API bei ihrer Registrierung hinterlegt hat. Diese Prüfung erfolgt bereits zum Zeitpunkt der Erstellung/Änderung der Policy, nicht erst zur Laufzeit.

Das Feld **exceptions** ist ausschließlich bei DENY-Policies semantisch gültig. Bei PERMIT-Policies wird es ignoriert bzw. darf nicht befüllt werden.

```
Policy
 ├── api id        : String          # Zugeordnete API-Ressource
 ├── effect        : Enum { PERMIT, DENY }
 ├── scopes          : List<Scopes>          # Liste von Scopes die im PERMIT-Fall freigegeben werden
 ├── conditions      : List<Condition>     # UND-verknüpft; alle  müssen zutreffen damit die Policy als Treffer
gilt
 └── exceptions      : List<Exception>    # Nur bei effect=DENY; ODER-verknüpft: trifft mindestens eine
Exception zu,
                                          # wird das DENY der Policy aufgehoben
```

Die formale Definition der Policy-Datenstruktur lautet wie folgt (in BNF-Notation):

```
<policy> ::= <permit-policy> | <deny-policy>

<permit-policy>      ::= "policy_id"      ":" <uuid>
                         "api_id"      ":" <api-ref>
                         "effect"      ":" "PERMIT"
                         "scopes"      ":" <scope-list>
                         "conditions"  ":" <condition-list>

<deny-policy>        ::= "policy_id"   ":" <uuid>
                         "api_id"      ":" <api-ref>
                         "effect"      ":" "DENY"
                         "conditions"  ":" <condition-list>
                         "exceptions"  ":" <exception-list>

<scope-list> ::= "[" <SCOPE> { "," <SCOPE> } "]"

<condition-list> ::= "[" "]" | "[" <CONDITION> { "," <CONDITION> } "]"

<exception-list> ::= "[" "]" | "[" <CONDITION> { "," <CONDITION> } "]"
```

```
Erweiterung - optionale LoA-Bedingung:

CONDITION := <ATTRIBUTE> <OPERATOR> <VALUE> <loa-constraint>

<loa-constraint> := "min_loa" <loa-level>
<loa-level> := "LOA_1" | "LOA_2" | "LOA_.."
<loa-constraint> ::= "[" "min_loa" ":" <SP> <loa-level> "]"
<loa-level> ::= "LOA_1" | "LOA_2" | "LOA_.."
```

Die Auswirkung (Effect) der jeweiligen Policy entscheidet darüber, ob Anfragen, die als Übereinstimmung mit den Kriterien der Policy bewertet werden, zugelassen oder abgewiesen werden sollen. Das Element **Policy Effect** kann folgende Ausprägungen haben:

- **PERMIT** *(erlauben)*: Es besteht eine Berechtigung, die Anfrage ist zulässig
- **DENY** *(ablehnen)*: Es besteht explizit KEINE Berechtigung, die Anfrage ist zu verweigern.

Berechtigungen auf APIs müssen explizit durch Policies erteilt werden.

Mehrere Policies können derselben API zugeordnet sein; ihre Ergebnisse werden durch den Combining Algorithmus bei der Auswertung kombiniert (siehe unten).

Das Datenmodell für **Policy Conditions** wird im nachfolgenden Abschnitt beschrieben, es umfasst die Elemente, um Selektionskriteren für Subjekte zu definieren, für die eine Policy gelten soll.

Policy Exceptions sind eine Liste von null oder mehreren Gruppen von Exceptions, wobei eine Exception ein oder mehrere **Conditions** enthält:

```
Exception
└── conditions      : List<Condition>   # UND-verknüpft: alle Conditions dieser
                                        # Exception müssen zutreffen, damit die
                                        # Exception als erfüllt gilt.
```

Policy Exceptions werden im Modell verwendet, um den Effekt von DENY-Policies zu steuern: DENY-Policies schließen den Zugriff für eine definierte Menge von Clients grundsätzlich aus. Um innerhalb einer solchen Ausschlussregel gezielte Ausnahmen zu ermöglichen, ohne eine separate PERMIT-Policy anlegen zu müssen, können DENY-Policies mit einer Ausnahmeliste (Exception) versehen werden. Eine Exception definiert dabei eine Teilmenge von Subjekten, für die die übergeordnete DENY-Policy nicht greift. Die Ausnahme wird – analog zur Policy selbst – attributbasiert formuliert: Subjekte, deren Attribute die Bedingungen der Exception erfüllen, werden von der DENY-Regel ausgenommen und fallen zurück in die weitere Policy-Auswertung.

Exceptions ermöglichen es, eng gefasste Sonderregelungen direkt an eine DENY-Policy zu knüpfen, ohne die Gesamtstruktur des Regelwerks durch zusätzliche Policies zu verkomplizieren. Sie sind dabei an die übergeordnete DENY-Policy gebunden und entfalten keine eigenständige Wirkung außerhalb dieses Kontexts.

Policy Conditions / Bedingung

Policy-Conditions (Bedingungen) bilden atomare, prüfbare Kriterien ab, denen ein Berechtigungs-Subjekt zum Zeitpunkt der Zugriffsprüfung genügen muss.

Bedingungen können beispielsweise Eigenschaften des API Consumer Clients sein, die im Föderalen Platformdirectory dem Client zugeordnet sind; wie beispielsweise dessen Betriebsorganisation, Behördenkennzeichen oder andere Attribute.

Das Datenmodell von Policy-Conditions wird formal wie folgt definiert (BNF):

```
CONDITION := <ATTRIBUTE> <OPERATOR> <VALUE>

ATTRIBUTE := <SUBJECT> | <CONTEXT>

SUBJECT := <SUBJECT-ATTRIBUTE>
CONTEXT  := <CONTEXT-ATTRIBUTE>

OPERATOR  := { EXISTS | EQ | NEQ | GT | GTE | LT | LTE | ... }
```

```
Erweiterung:

CONDITION := <ATTRIBUTE> <OPERATOR> <VALUE> <loa-constraint>

<loa-constraint> := "min_loa" <loa-level>
<loa-level> := "LOA_1" | "LOA_2" | "LOA_3" | "LOA_.."
<loa-constraint> ::= "[" "min_loa" ":" <SP> <loa-level> "]"
<loa-level> ::= "LOA_1" | "LOA_2" | "LOA_3" | "LOA_4"
```

Bedingungen bestehen immer aus 3 Elementen und bilden einen Term für die Vergleichsoperation:

- **<ATTRIBUTE>** steht für ein Attribut des Subjects (z. B. des API Client) oder eine Kontextvariable (z. B. „aktuelle Uhrzeit")
- Es können alle Attribute verwendet werden, die im Attribut-Katalog der Berechtigungsverwaltung des FöPD verfügbar sind.
- **<OPERATOR>** steht für die Vergleichsoperation der Bedingung,
- **<VALUE>** steht für den Bedingungswert, den das Attribut nach Anwendung des Vergleichsoperators erfüllen muss.

Erweiterung

- **<loa-contraint>** steht für den mindestens erforderlichen Grad der Vertrauenswürdigkeit des Attributswerts in der Regel (Level-of-Assurance)

Für das Policy-Modell gelten folgende semantische Eigenschaften:

- Reihenfolge der Conditions ist semantisch neutral. Innerhalb einer Policy oder Exception beeinflusst die Reihenfolge der Conditions nicht das Ergebnis.
- Exceptions sind ODER-verknüpft. Trifft mindestens eine Exception zu, wird das DENY aufgehoben. Jede Exception ist ein eigenständiger Ausnahmetatbestand.

- Exceptions berühren keine anderen Policies. Eine Exception in Policy A hat keinen Einfluss auf eine andere DENY-Policy B, die dieselbe API schützt. Jede Policy wird vollständig isoliert ausgewertet.
- Custom-Attribute. Eine Condition darf "`custom.<api_id>.*`"-Attribute nur referenzieren, wenn `<api_id>` mit der `api_id` der eigenen Policy übereinstimmt.

Für die Beschreibung des nachfolgenden Prüfschema wird definiert, dass die Auswertung einer Policy-Condition durch folgende folgende Funktions-Signatur initiiert wird:

```
eval_condition (condition: Condition, AV: AttributeValueSet) → Boolean
```

wobei "`AV: AttributeValueSet`" die Menge der Attributwerte des anfragenden Clients bzw. der Kontextattribute ist.

Die Auswertung erfolgt in folgenden Schritten:

Schritt 1 – LOA-Prüfung (nur wenn `condition.min_loa` gesetzt)

```
attr = AV[condition.subject]  # hole Attributwert des Client/Kontext

WENN attr fehlt     → FALSE   # gefordertes Attribut fehlt
WENN attr.loa < condition.min_loa → FALSE   # Vertrauensgrad unzureichend
```

Wenn die Regel einen Mindestwert für das Level-of-Assurance (`min_loa`) für den Attributwert gesetzt hat, wird zunächst geprüft, ob das Subjekt über das geforderte Attribut verfügt. Ist das Attribut nicht vorhanden, wird die Regel mit FALSE bewertet.

Ebenfalls mit FALSE bewertet wird die Regel, wenn das Attribut zwar vorhanden ist, der LoA-Wert des Attributs aber nicht dem geforderten LoA-Wert der Policy-Regel entspricht.

Enthält die Policy-Bedingung keinen Mindest-LoA, wird die Prüfung regulär fortgesetzt:

Schritt 2 – Wertvergleich:

```
WENN attr = AV[condition.subject] ∈ { EXISTS, NOT_EXISTS }:  ## attr fehlt
    EXISTS     → attr vorhanden   ? TRUE : FALSE
    NOT_EXISTS → attr vorhanden   ? FALSE : TRUE

attr = AV[condition.subject]      # hole Attributwert des Client/Kontext
eval_operator(attr.value, c.operator, c.value)      → TRUE | FALSE        #
Operator, Attributwert und Condition auswerten
```

Wenn ein in der Policy-Condition gefordertes Attribut zum Zeitpunkt der Auswertung fehlt, liefert die Auswertung immer FALSE. Das Nichtvorhandensein eines für die Regel erforderlichen Attributs ist damit semantisch gleichbedeutend mit einer nicht erfüllten Bedingung.

Die nachfolgenden (nicht-normativen) Beispiele illustrieren mögliche Policy-Conditions:

#	Bedingung	Bedeutung
1	`acting_for.behoerde.typ EQ Melderegister`	Wahr, wenn der Behördentyp "Melderegister" ist
2	`acting_for.behoerde.typ IN [Melderegister, Polizei]`	Wahr, wenn der Behördentyp "Melderegister" oder „Polizei“ ist
3	`software_statement.id = urn:platform-directory:ss:a1b2c3d4-e5f6-...`	Wahr, wenn der Client das Software-Statement mit der genannten ID repräsentiert (Einzelberechtigung)

LOA in Berechtigungsregeln

API-Berechtigungen können in ihren Bedingungen einen Mindest-LoA für referenzierte Attribute fordern. Damit wird in der Regelauswertung nicht nur geprüft, ob ein Attribut einen bestimmten Wert hat, sondern auch, ob dieser Wert hinreichend vertrauenswürdig ist.

LOA erlaubt differenzierte Sicherheitsanforderungen auf Attributebene, ohne separate Berechtigungsklassen einführen zu müssen.

Beispiel einer LOA-Bedingung:

Bedingung:	*client.behoerde_typ EQ Melderegister [min_loa: LOA_3]*
Bedeutung:	Wahr, wenn der Behördentyp "Melderegister" ist UND dieser Wert durch eine autorisierte Stelle bestätigt wurde

[Lesehinweis: Das LoA-Konzept befindet sich noch in Überarbeitung ist nur vorbehaltlich Teil des Konzepts.]

Policy Regelauswertung / Combining Algorithmus

Zur Auswertung einer Zugriffsanfrage auf ein API werden folgende Inputs verwendet:

- **Client-Kontext:** Die validierten Attribute des anfragenden Clients aus dem Platform-Directory (aus dem Software Statement, Metadaten).
- **API-Kontext:** Identifikation der angefragten API (`api_id`).
- **Policies:** Alle aktiven Policies, die der `api_id` zugeordnet sind.

Die Auswertung erfolgt in zwei Phasen:

1. zunächst werden alle Policies individuell ausgewertet (eine Vorauswahl auf das jeweilige Resource API ist zulässig)
2. anschließend werden ihre Ergebnisse durch den **Combining Algorithm** zum Gesamtergebnis kombiniert.

Für die nachfolgende schematische Beschreibung der Auswertungslogik wird die Auswertung von Policies mit der folgenden Signatur inittiert:

```
eval_policy(p: Policy, AV: AttributeValueSet) → PolicyResult { PERMIT | DENY | NO_MATCH }
```

Eine Policy aggregiert ihre Conditions mit **strikter Konjunktion (logisches UND)**:

```
conditions_match = ∀ c ∈ p.conditions : eval_condition(c, A) = TRUE
```

Alle Policy-Conditions einer Policy müssen als wahr ausgewertet werden, damit die Policy übereinstimmt und zur Entscheidung beiträgt.

Bei der Auswertung kann die Konjunktion mittels Short-Circuit-Evaluation erfolgen: Sobald eine Condition FALSE liefert, wird die Auswertung abgebrochen.

```
WENN conditions_match = FALSE → NO_MATCH    # Policy trifft nicht zu
```

Policy Exceptions dienen dazu, den Effekt von DENY-Policies aufzuheben. Eine Policy kann null oder mehrere **Policy Exceptions** beinhalten, wobei eine Exception ein oder mehrere Conditions umfasst.

Alle Conditions einer Exception müssen zutreffen, damit die Exception zutrifft (UND-Verknüpfung). Enthält eine Policy mehrere Exceptions, reicht eine zutreffende Exception damit die Policy Exception zutrifft wirksam ist (ODER-Verknüpfung).

Die Verknüpfungslogik über beide Ebenen zusammengefasst:

```
exceptions_match =
    ∃  e ∈ p.exceptions :      ← ODER: mindestens eine Exception muss zutreffen
        ∀  c ∈ e.conditions :  ← UND:  alle Conditions der Exception müssen erfüllt sein
            eval_condition(c, A) = TRUE
```

Treffen für eine Resource API-Anfrage mehrere Policies zu, werden diese gemäß der folgenden Logik miteinander kombiniert:

1. Wenn genau null Policies zu „**PERMIT**“ führen, lautet das Gesamtergebnis der Auswertung „**DENY**“.
2. Wenn mindestens eine übereinstimmende Policy zu „**PERMIT**“ führt und genau null Policies zu „**DENY**“ führen, lautet das Gesamtergebnis der Auswertung „**PERMIT**“.

3 Wenn mindestens eine übereinstimmende Policy zu „**DENY**“ führt, werden die EXCEPTIONS der DENY-Policies ausgewertet:

   1 Treffen die Bindungen mindestens einer Exception der Policy zu, wir das DENY der Policy aufgehoben.

   2 Trifft keine Exception der Policy zu, gilt das DENY der Policy.

Nach Auswertung aller Policies werden die verbleibenden PERMIT-Policies ausgewertet:

4 Für alle verbleibenen „**PERMIT**“ Policies werden die per Policy freigegebene Scopes aggregiert;

   1 wurden in der PDP-Anfrage Scopes angefordert, werden ausschließlich jene Scopes zurückgeliefert, die sowohl vom Client angefragt als auch durch mindestens eine PERMIT-Policy freigegeben wurden (Schnittmenge).

   2 Ist die Schnittmenge leer, ändert sich das Gesamtergebnis zu DENY.

```
nur wenn conditions_match = TRUE:

WENN p.effect = PERMIT:
    → PERMIT

WENN p.effect = DENY:
    # Exceptions prüfen (ODER-verknüpft)
    FÜR JEDE exception ∈ p.exceptions:
        exception_match = ∀ c ∈ exception.conditions : eval_condition(c, A)
= TRUE
        WENN exception_match = TRUE:
            → NO_MATCH    # Exception greift: DENY wird durch NO_MATCH der
Policy aufgehoben

    → DENY    # Keine Exception zugetroffen
```

Kernprinzipien des Combining Algorithm

Mit diesem Schema werden folgende Kernprinzipien in der Berechtigungslogik umgesetzt:

- **Implicit Deny:** Trifft keine Policy zu, ist das Standardergebnis DENY. Berechtigungen müssen vollständig und explizit durch PERMIT-Policies erteilt werden.

- **DENY-Dominanz:** Eine einzige zutreffende DENY-Policy überstimmt beliebig viele zutreffende PERMIT-Policies. Die Reihenfolge, in der Policies ausgewertet werden, ist semantisch irrelevant – jede Policy wird vollständig und unabhängig ausgewertet.
- **Positionsunabhängigkeit:** Die Reihenfolge der Policies hat keine Auswirkung auf das Ergebnis.
- **Exceptions begrenzen DENY-Dominanz lokal:** Exceptions können das DENY einer einzelnen Policy für einen bestimmten Nutzer-Teilraum aufheben – aber ausschließlich innerhalb dieser Policy. Sie beeinflussen nicht den Combining Algorithm und nicht das Ergebnis anderer Policies.
- **Scope-Union über alle PERMIT-Matches:** Der policy-erlaubte Scope-Satz ist die Vereinigung der Scopes aller zutreffenden PERMIT-Policies. Ein Client, der mehrere PERMIT-Policies erfüllt, erhält die Gesamtmenge der darin freigegebenen Scopes – begrenzt durch seine eigene Anfrage.
- **Scope-Intersection mit Client-Anfrage (Least Privilege):** Der tatsächlich vergebene Scope-Satz ist die Schnittmenge aus policy-erlaubten und client-beantragten Scopes. Ein Client erhält nicht mehr Scopes, als er beantragt hat. Hierdurch wird dem Principle of Least Privilege Rechnung getragen: Clients deklarieren zur Laufzeit explizit, welche Scope sie nutzen möchten.
- **Scope-Hoheit beim API-Betreiber:** Policies können ausschließlich Scopes referenzieren, die das API zuvor registriert hat. Eine Policy kann den Berechtigungsrahmen des APIs nicht überschreiten. Der Scope-Namensraum ist per API isoliert – Kollisionen zwischen verschiedenen APIs sind strukturell ausgeschlossen.

Beispiele für die Wirkweise des Combining Algorithm

Beispiel 1: Implicit Deny

Resource API	Policy	Policy Effect	Policy trifft auf Anfrage zu	Exceptions trifft auf Anfrage zu	Scopes
Beispiel-1	P1	PERMIT	Nein	Nein	Lesen, Schreiben
Beispiel-1	P2	PERMIT	Nein	Nein	Lesen
Beispiel-1	P3	DENY	Nein	Nein	-

Ergebnis: DENY

Scopes freigegeben: Keine

Erläuterung: Keine Policy trifft zu (Implicit Deny)

Beispiel 2: Permit, mehrere Policies

Resource API	Policy	Policy Effect	Policy trifft auf Anfrage zu	Exceptions trifft auf Anfrage zu	Scopes
Beispiel-2	P1	PERMIT	Ja	Nein	Lesen, Schreiben
Beispiel-2	P2	PERMIT	Ja	Nein	Lesen
Beispiel-2	P3	DENY	Nein	Nein	-

Ergebnis: PERMIT

Scopes freigegeben: Lesen, Schreiben

Erläuterung: Berechtigung durch P1 + P2 gewährt, P3 trifft nicht zu.

Beispiel 3: Explicit Deny

Resource API	Policy	Policy Effect	Policy trifft auf Anfrage zu	Exceptions trifft auf Anfrage zu	Scopes
Beispiel-3	P1	PERMIT	Ja	Nein	Lesen, Schreiben
Beispiel-3	P2	PERMIT	Ja	Nein	Lesen
Beispiel-3	P3	DENY	Ja	Nein	-

Ergebnis: DENY

Scopes freigegeben: Keine

Erläuterung: Ablehnung durch P3, keine Exceptions in P3 machen P3 unwirksam.

Beispiel 4: Deny Exception

Resource API	Policy	Policy Effect	Policy trifft auf Anfrage zu	Exceptions trifft auf Anfrage zu	Scopes
Beispiel-4	P1	PERMIT	Ja	-	Lesen, Schreiben
Beispiel-4	P2	PERMIT	Ja	-	Lesen
Beispiel-4	P3	DENY	Nein	Nein	-
Beispiel-4	P4	DENY	Ja	Ja	-

Ergebnis: PERMIT

Scopes freigegeben: Lesen, Schreiben

Erläuterung: P1 + P2 liefern PERMIT, P4 trifft zu, aber zutreffende Exception in P4 hebt Gültigkeit von P4 auf.

5.1.3.3 Attributkatalog

Der Attributkatalog ist ein normativer, zentral verwalteter Bestandteil des Föderalen Platform-Directory. Er definiert verbindlich alle offiziellen Attribute, die in Berechtigungsregeln referenziert werden dürfen, und verhindert durch ein striktes Namensraum- und Scoping-Modell den unkontrollierten Wildwuchs von Attributnamen.

Der Attributkatalog verwaltet den Bestand verfügbarer Attribute, sowie deren normative Definition.

Konkrete Zuordnungen von Attributen auf Subjekte und Ressourcen (Software-Statements resp. APIs) der Berechtigungsverwaltung werden über die Attribute Authority des FöPD (FöPD-AA) realisiert.

Namensräume und Scoping

Attribute werden innerhalb eines **Namensraums (Namespace)** organisiert. Ein Attribut ist dabei jeweils genau einem von mehreren möglichen Namensräumen zugeordnet.

Zur Referenzierung von Attributen verwendet dieses Konzept die Dot-Notation (x.y.z), bei der jedes durch einen Punkt getrennte Segment eine hierarchische Ebene im Namensraum beschreibt. Diese Notation ermöglicht eine konsistente und skalierbare Organisation der Attribute im Attribut-Katalog.

Für die Organisation der Namespaces auf oberster Ebene gilt folgende Festlegung:

Tabelle 34: Namesraumspezifikation

Namens-raum	Präfix	Verwaltung	Verwendung
Offiziell	`software.*` `api.*`	Zentral durch Katalog-Governance	In allen Policies referenzierbar
Custom	`custom.<api_id>.*`	Durch den jeweiligen API-Betreiber	Nur in der Policy der eigenen API

Custom-Attribute sind explizit auf den Namensraum der registrierenden API gescopet. Sie können offizielle Attribute weder überschreiben noch überlagern. Ein Attribut wie `custom.api-xyz.interne_kategorie` ist für andere APIs semantisch nicht sichtbar und in deren Policies nicht referenzierbar.

Attributdefinition im Katalog

Für die Verwaltung des Attributkatalogs sind mindestens folgende Meta-Eigenschaften erforderlich:

Tabelle 35: Attributskatalog mit Meta-Eigenschaften

Eigenschaft	Typ	Beschreibung
`attribute_id`	String	Kanonischer Name, z. B. "`client.behoerde_typ`"Segoe UI
`namespace`	Enum { `OFFICIAL, CUSTOM` }	
`scope`	String	Bei `CUSTOM: api_id` des Eigentümers
`data_type`	Enum { `STRING, INTEGER, DATE, BOOLEAN, STRING_LIST` }	Typ des Attributes
`allowed_values`	List <`data_type`>	Optional: kontrolliertes Vokabular
`description`	String	Menschenlesbare Bedeutungsbeschreibung
`status`	Enum { `ACTIVE, DEPRECATED, INACTIVE, PROPOSED` }	Gibt an, ob das Attribut für Polices verwendet werden kann.
`version`	SemVer	Version des Attributs
`loa_min`	integer	[Optional] Mindest-LOA, mit dem dieses Attribut befüllt sein muss

Offizielle Standard-Attribute des Katalogs

Die nachfolgenden Attribute werden automatisch durch die Systemverwaltung bereitgestellt und als Standardattribute im Katalog definiert.

Tabelle 36: Standardattribute des Katalogs

Attribut	Namensraum	Typ	Beschreibung
`software.locked`	Official	boolean	Sperrkennzeichen des Clients
`context.*`	Official	<tbd>	Virtuelle Attribute die in Policies verwendet werden, um einen Zustand zur Laufzeit abzubilden, z. B. „Uhrzeit“
<tbd>			[weitere Attribute werden in einer späteren Version des Konzeptes formalisiert]

Die tatsächlich im Katalog verfügbaren Attribute richten sich nach dem operativen und administrativen Bedarf. Bevorzugt werden bereits verfügbare Attributkataloge und Attributquellen aus der Verwaltung bedarfsmäßig integriert, wie beispielsweise https://fimportal.de/kataloge.

Offene Punkte

[Lesehinweis: Zum aktuellen Stand der Konzeption bedürfen die folgende Aspekte weiter Konkretisierung. Dies erfolgt in den kommenden Iterationen des Konzepts]

- Definition der Syntax und internen Darstellung der Attribute im Attributkatalog, in der Verwendung der Policy-Definition sowie in der Kommunikationsschicht (AuthZEN, s. u.)
- Definition der semantisch-fachlichen Bedeutung der Standard-Attribute des Attribut-Katalogs
- Anforderungen an die organisatorischen Aspekte der Attribut-Governance

Level of Assurance (LoA)

[Anmerkung: Das vorliegende Unterkapitel befindet sich noch in konzeptioneller Überarbeitung und ist nur vorbehaltlich Teil des Konzepts.]

Attributwerte, die Platform-Directory für einen Client oder eine API hinterlegt sind, oder zum Laufzeitkontext einem Authorization Server vorliegen, können einen Level of Assurance (LoA) tragen.

Der LoA drückt aus, wie vertrauenswürdig die Herkunft und Validierung dieses Attributwerts ist – unabhängig vom Wert selbst.

Ein LoA-Wert kann in Policies verwendet werden, um für besonders sensible Zugriffsbereiche erhöhte Anforderungen an die Vertrauenswürdigkeit der zugreifenden Subjekte zu stellen.

Folgende Tabelle legt die Bedeutung der LoA-Stufen dar:

Tabelle 37: Stufen der Level of Assurance (LoA)

Stufe	Bezeichnung	Bedeutung
LOA_1	Selbstauskunft	Vom Client selbst angegeben, nicht verifiziert
LOA_2	Bestätigt	Durch eine autorisierte Stelle geprüft und bestätigt
LOA_3	Zertifiziert	Durch formalen Akt nachgewiesen (z. B. amtliche Registrierung, PKI-Zertifikat)
LOA_...		

Anmerkung: Die Definition, Vergabe und Verwaltung der LoA-Attribute obliegt einem Governance-Gremium und ist nicht Gegenstand dieses Konzepts. (vgl. https://csrc.nist.gov/glossary/term/level_of_assurance)

LoA-Werte für APIs bzw. Software-Client, die im Platformdirectory gespeichert werden als Metadatum am Attributwert im FöPD selbst gespeichert:

```
AttributeValue
├── value          : Any              # Der eigentliche Attributwert
├── loa            : Enum { LOA_1, LOA_2, LOA_3, LOA_.. }
├── validated_by   : String | null    # Kennung der prüfenden Stelle
└── validated_at   : DateTime | null  # Zeitpunkt der Validierung
```

5.1.4 Systemkomponenten & Kommunikation

5.1.4.1 Zentrale Komponenten

Die zentrale Systemschicht der Berechtigungssteuerung umfasst folgende Komponenten, die gemeinsam die Grundlage für die plattformweite Berechtigungssteuerung bilden:

- Der **Policy Administration Point (PAP)** ist die zentrale Verwaltungsschnittstelle für Berechtigungsregeln. Er bildet die einzige autorisierte Schreibschnittstelle gegenüber dem Policy Store und stellt sicher, dass Policies ausschließlich über kontrollierte, nachvollziehbare Prozesse erstellt, bearbeitet oder gelöscht werden können.

- Der **Policy Information Point (PIP)** ist das Persistenzmedium für Attributdaten. Er bezieht Attributwerte aus dem Platform Directory und stellt diese zur Laufzeit dem PDP zur Verfügung.
- Der **Policy Store** ist das Persistenzmedium für alle Policies der Plattform. Er verantwortet die Versionierung des Regelbestands. Der **Policy Retrieval Point (PRP)** ist für die selektive Distribution von Policies an die zuständigen PDP-Instanzen der Basisdienste verantwortlich.
- Der **Attributkatalog** definiert alle im Berechtigungskontext zulässigen Attribute und deren Werteformate. Er ist die normative Grundlage für die Formulierung von Policies und stellt sicher, dass berechtigungsrelevante Attribute plattformweit einheitlich und konsistent verwendet werden.

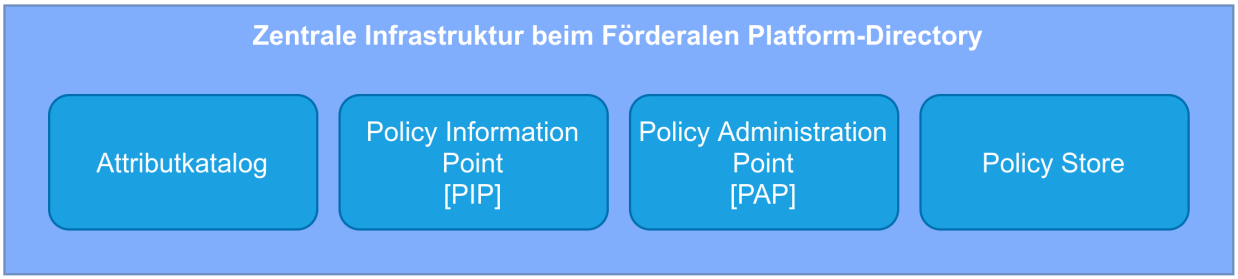


Abbildung 59: Zentrale Berechtigungsinfrastruktur des Föderalen Platform-Directories

Policy Administration Point (PAP)

Der **Policy Administration Point** des FöPD ist die zentrale Verwaltungsschnittstelle für alle Berechtigungsregeln der Plattform. Er bildet die einzige autorisierte Schreibschnittstelle gegenüber dem Policy Store und stellt sicher, dass Policies ausschließlich über kontrollierte, nachvollziehbare Prozesse erstellt, bearbeitet oder gelöscht werden können.

Der PAP stellt zwei Zugangswege bereit:

- Über eine Benutzeroberfläche können Betreiber Policies ohne tiefe technische Kenntnisse verwalten; die Oberfläche führt den Nutzer durch den Erstellungsprozess, bietet geführte Eingabemasken sowie eine Vorschau auf die Auswirkungen einer Policy.
- Ergänzend steht eine API zur Verfügung, über die Policies automatisiert verwaltet oder in CI/CD-Pipelines integriert werden können.

Validierung

Bevor eine über den PAP eingereichte Policy in den Policy Store übernommen wird, durchläuft sie eine Validierungsphase. Der PAP prüft:

- syntaktische Korrektheit der Policy,
- referenzielle Integrität der verwendeten Attribute gegenüber dem Attributkatalog,
- sowie die Zuständigkeit des einreichenden Betreibers für das betroffene API.

Erst nach erfolgreicher Validierung wird die Policy zur Persistierung an den Policy Store übergeben und von dort in den Distributions- und Versionierungsprozess überführt.

PDP-Registrierung und Vertrauenssteuerung

Über den PAP werden die Registrierung und Zuordnung von PDP-Instanzen der jeweiligen Basisdienst-Betreiber gesteuert. Dabei werden folgende Aspekte verwaltet:

- Registrierung vertrauenswürdiger PDPs im Platform Directory mit eindeutiger Zuordnung zum jeweiligen Betreiber,
- Pflege der Zuordnung zwischen Betreiber, API und zuständiger PDP-Instanz,
- Sicherstellung, dass Policy-Informationen ausschließlich an registrierte und als vertrauenswürdig eingestufte PDPs ausgeliefert werden.

Nicht registrierte oder nicht autorisierte PDP-Instanzen erhalten keinen Zugriff auf die im Policy Store hinterlegten Policies. Für die Auslieferung der Policies und Kommunikation zu den dezentralen PDPs ist der PRP als dedizierte Komponente zuständig. So wird einerseits sichergestellt, dass Policy-Informationen und Metadaten nur an berechtigte PDPs ausgeliefert werden. Weiterhin wird durch die Entkopplung zwischen PAP und PRP sichergestellt, dass die Angriffsfläche für exponierte Komponenten minimiert wird und sich etwaige Kompromittierungen im Bereich der Policy-Auslieferung nicht auf die Administrationsschicht auswirken.

Health-Monitoring und Metriken

Der PAP/PRP stellt grundlegende Monitoring- und Healthfunktionen bereit, die einen Überblick über den Zustand der Policy-Administration ermöglichen. Dazu gehören insbesondere:

- Betriebsstatus und Verfügbarkeit des PAP selbst,

- Metadaten und Metriken zur Policy-Distribution, etwa Abrufzeitstempel und ausgelieferte Versionen je PDP-Instanz,
- Nachvollziehbarkeit, ob registrierte PDPs Policies erfolgreich abgerufen haben und auf welchem Versionsstand sie sich befinden,
- Grundlegende Kennzahlen zur Administrationsaktivität, etwa Anzahl aktiver Policies je Betreiber oder Häufigkeit von Regeländerungen.

Diese Funktionen dienen in erster Linie der Plattformadministration und unterstützen die frühzeitige Erkennung von Problemen in der Policy-Auslieferung oder im Betrieb angebundener PDPs.

Internes Berechtigungsmodell

Der PAP selbst wird über ein eigenes Berechtigungsmodell verfügen, das regelt, welche Nutzer welche Aktionen im PAP ausführen dürfen. Dieses interne Berechtigungsmodell ist nicht Gegenstand des vorliegenden Konzepts; es wird in einem gesonderten Implementierungskonzept definiert und richtet sich nach den fachlichen und organisatorischen Anforderungen der Plattformadministration.

Grundsätzlich werden Nutzergruppen mit unterschiedlichen Rechten und Sichtweiten bedient:

- API-Betreiber arbeiten im Self-Service-Modus: Sie können Policies ausschließlich für die APIs verwalten, für die sie im Platform Directory als verantwortliche Stelle registriert sind.
- Governance- und Administrationsstellen verfügen über eine plattformweite Sicht und erweiterte Berechtigungen.

Policy Information Point (PIP)

Der **Policy Information Point** ist die Informationsversorgungskomponente der Berechtigungsarchitektur: Seine Aufgabe besteht darin, aktuelle und konsistente Attributdaten für die Policy-Persistierung bereitzustellen. Der PIP abstrahiert den Zugriff auf unterschiedliche Attributquellen und stellt dem Policy Store eine einheitliche Abfrageschnittstelle zur Verfügung. Der PIP hat keine Laufzeit-Interaktion mit dem AS – er wirkt ausschließlich in der Persistierungs- und Verteilungsschicht.

Die primäre Attributquelle des PIP ist das zentrale Platform Directory. Von dort bezieht der PIP die Attribute der Software Statements aller registrierten Clients und APIs – darunter technische

Identifier, organisatorische Zuordnungen (wie z. B. Behördenkennungen, Sachzuständigkeiten, u. a. Klassifizierungsmerkmale) die eine Unterscheidung der Berechtigungssubjekte ermöglichen.

Die Aufgaben des PIP im Überblick:

- Bereitstellung und Auslieferung von Metadaten über Clients (Software Statements) mit Attributen an relevante PDPs
- Berechnung der **Relevanz**: welche PDPs welche Software Statements benötigen

Was der PIP ausdrücklich nicht tut:

- Er berechnet keine Berechtigungsentscheidungen vor – das ist Aufgabe des lokalen PDP zur Laufzeit
- Er liefert keine Policies aus - das ist Aufgabe des PRP/PAP
- Er kommuniziert nicht mit dem AS – alle relevanten Metadaten werden als eigener Datenbestand (Facts) mit den Policies ausgeliefert
- Er liefert keine vollständigen Inhalte des Platform Directory aus (Datensparsamkeit als strukturelles Prinzip)

Die folgende Abbildung illustriert die Integration des Policy Information Point in die Gesamtarchitektur:

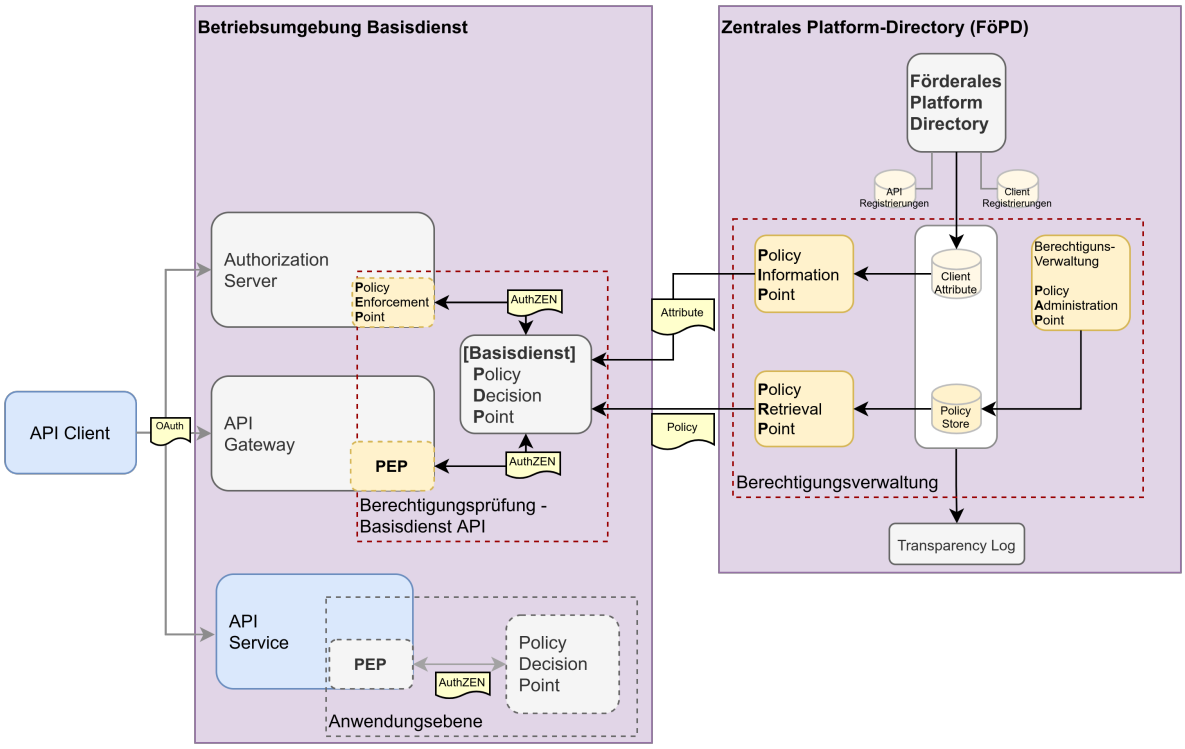


Abbildung 60: Integration des Policy Information Point

Die Kommunikation zwischen PIP und PDP erfolgt verschlüsselt und authentisiert. Nur authentisierte PDPs erhalten Datenbestände mit Attributinformationen. Eine interne Berechtigungs-Policy stellt dabei sicher, dass der PIP nur die Informationen ausliefert, die ein PDP zur Berechtigungsprüfung benötigt.

Vom PIP ausgelieferten Datenpakete an PDPs werden vom PIP kryptographisch signiert und mit einem Versionsschlüssel versehen. Versionskennung und Signatur wird vom PIP in das Transparency-Log protokolliert. Auf diesen Weg können PDPs die Integrität und Authentizität der übermittelten Attribute über einen zweiten Kommunikationspfad verifizieren.

Policy Store, Policy Retrieval Point (PRP)

Der Policy Store ist das zentrale Persistenzmedium der Plattform für Berechtigungsregeln. Er ist die einzige autoritative Datenquelle für den Bestand gültiger Policies und bildet das Bindeglied zwischen der administrativen Seite der Plattform (PAP) und der operativen Berechtigungsprüfung (PDP).

Schreibzugriffe auf den Policy Store sind ausschließlich über den PAP möglich; der PDP sowie weitere Konsumenten haben ausschließlich lesenden Zugriff.

Die folgende Abbildung zeigt die logische Sicht auf den Policy Store:

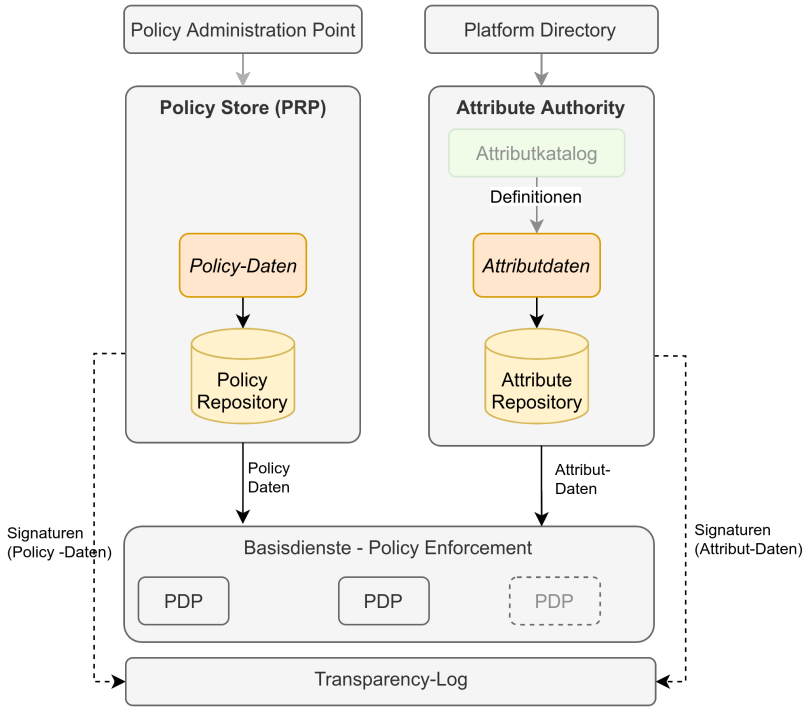


Abbildung 61: Logische Sicht auf den Policy Store

Persistierung von Policies

Jede über den PAP erstellte und erfolgreich validierte Policy wird im Policy Store dauerhaft abgelegt. Der Policy Store enthält damit:

- die abstrakte Regellogik in maschinenlesbarer Form,
- die in der Regellogik als Kriterien referenzierten Attribute,

Policies und die zugehörigen Attributdaten werden dabei prinzipiell unabhängig voneinander betrachtet: Jede Policy referenziert Attribute ausschließlich über die im Attributkatalog definierten Bezeichner, sodass Attributwerte und Regellogik unabhängig voneinander aktualisiert werden können.

Diese Trennung hat einen wesentlichen operativen Vorteil: Policies und Attributdaten können unabhängig voneinander an die PDPs ausgeliefert werden. Kurzfristige Änderungen an

Attributwerten – etwa durch Aktualisierungen im Platform Directory – werden unmittelbar und ohne ein vollständiges Policy-Update an die betroffenen PDPs transportiert.

Versionierung und Nachvollziehbarkeit

Jede Änderung an einer Policy erzeugt einen neuen versionierten Eintrag im Policy Store. Neuanlage, Bearbeitung und Löschung werden gleichermaßen erfasst. Zu jeder Version werden folgende Metadaten dauerhaft protokolliert:

- Zeitpunkt der Änderung,
- Stelle oder Person, die die Änderung durchgeführt hat
- Art der Änderung (Neuanlage, Aktualisierung, Deaktivierung, Löschung),
- vollständiger Regelinhalt der jeweiligen Version.

Der vollständige Änderungsverlauf bleibt dauerhaft für Auditierungszwecke erhalten. Ältere Versionen können bei Bedarf wiederhergestellt werden, sodass fehlerhafte Regeländerungen operativ korrigiert werden können, ohne manuelle Rekonstruktion der Ausgangslage.

Distribution an dezentrale Basisdienste-PDPs - Policy Retrieval Point (PRP)

Über den Policy Store wird gesteuert, welche Policies an welche PDP-Instanz ausgeliefert werden. Die Distribution erfolgt auf Basis der im PAP hinterlegten Zuordnung zwischen Policies, APIs, Basisdiensten und registrierten PDP-Instanzen. Für jede Änderung im Policy Store wird ermittelt, welche PDP-Instanzen betroffen sind; diese erhalten die aktualisierten Policies zugestellt.

Jedes Policy-Update das an PDPs ausgeliefert wird, wird mit einer Versionskennung versehen und durch eine kryptographische Signatur authentisiert. Version und Signatur werden vom Policy Store im Transparency-Log veröffentlicht und an die PDPs ausgeliefert.

Die selektive Distribution stellt sicher, dass:

- jeder PDP ausschließlich die Policies erhält, die für seinen eigenen Zuständigkeitsbereich relevant sind,
- Berechtigungsregeln eines Basisdienstes nicht an PDP-Instanzen fremder Basisdienste ausgeliefert werden,

- ausschließlich registrierte und als vertrauenswürdig eingestufte PDPs Policies abrufen können,
- jeder PDP jederzeit einen vollständigen und konsistenten Regelbestand vorhält.

Neben der aktiven Zustellung können registrierte PDPs Policies und Attributdaten auch explizit abrufen, etwa beim initialen Start oder nach einer Unterbrechung der Verbindung. Der Policy Store liefert in diesem Fall den vollständigen, aktuellen Regelbestand für den jeweiligen Zuständigkeitsbereich aus.

Die Kommunikation zwischen PDPs und dem Policy Store erfolgt ausschließlich über ein sicheres Transportprotokoll (HTTPS/TLS). Die Authentisierung zwischen PDP und Policy Store wird über etablierte Standardprotokolle realisiert (OIDC, mit DPoP/mTLS) und stellt sicher, dass ausschließlich registrierte und authentisierte PDP-Instanzen Zugriff auf den Policy Store erhalten.

Konsistenz und Integrität

Der Policy Store gewährleistet Zeitpunkt die Konsistenz des gespeicherten Regelbestands. Konkurrierende Schreibzugriffe werden durch geeignete Mechanismen verhindert. Jede gespeicherte Policy ist mit einer eindeutigen Version und einem Zeitstempel versehen, anhand derer PDPs den Aktualitätsstand ihres lokalen Regelbestands mit dem Policy Store abgleichen können. Abweichungen können eine gezielte Nachsynchronisation auslösen, ohne dass der gesamte Regelbestand neu übertragen werden muss.

5.1.4.2 Dezentrale Komponenten

Abbildung 62: Dezentrale Berechtigungsinfrastruktur eines Basisdienstes

Authorization Server der Basisdienste

Der Authorization Server (AS) des Basisdienstes ist der lokale Vertrauensanker in der Infrastruktur des Basisdienstes.

Im Rahmen der Berechtigungsteuerung fungiert der AS als **Policy Enforcement Point (PEP):** er stellt Access Token für Clients nur dann aus, wenn der **Policy Decision Point (PDP)** des Basisdienstes dies aufgrund einer Policy-Regel entscheidet. Der AS trifft somit selbst keine inhaltlichen Berechtigungsentscheidungen.

Der Authorization Server übernimmt folgende Aufgaben:

- validiert das Software Statement des Clients gegenüber dem Platform-Directory (Registrierung & Authentisierung des Clients)
- Entgegennehmen von Client-Registrierungen über das Dynamic Client Registration Protocol (RFC 7591, siehe Kapitel 4.4.2.5)
- Entgegennehmen von Zugriffsanfragen von API-Clients auf Basisdienst-APIs
- Weiterleiten der Berechtigungsentscheidung an den PDP
- Ausstellen von Access Token für registrierte Clients aufgrund einer Policy-Freigabe durch den lokalen PDP

Authorization Server und Policy Decision Point kommunizieren über das AuthZEN-Protokoll miteinander (s.u.). Die Kommunikation wird über Standardprotokolle abgesichert und authentisiert (HTTP/TLS, OIDC, DPoP/mTLS).

API Gateway der Basisdienste

Das API Gateway des Basisdienstes ist für den Schutz der direkten Kommunikation zwischen API Client und API verantwortlich.

Im Kontext der Berechtigungssteuerung übernimmt das API Gateway folgende Aufgaben:

- kryptographische Validierung des Access Token des API Consumer Client – *ist das Token kryptographisch gültig?*
- Validierung der Zugriffslegitimation – *darf das Access Token noch verwendet werden?*

Analog zum Authorization Server des Basisdienst übernimmt das API-Gateway die Funktion eines Policy Enforcement Point (PEP) und stellt sicher, dass das API nur mit einem gültigen Access Token verwendet werden kann.

Während der AS in Zusammenarbeit mit dem PDP darüber entscheidet, ob die Kommunikation mit dem Basisdienst-API für diesen API Client überhaupt ermöglicht wird (diese Prüfung findet einmalig während des Token Exchange statt), prüft der API-Gateway sämtliche nachfolgende, laufende Kommunikation zwischen API Client und API ab.

API Gateway und Policy Decision Point kommunizieren über das AuthZEN-Protokoll miteinander (s. u.). Die Kommunikation wird über Standardprotokolle abgesichert und authentisiert (HTTP/TLS, OIDC, DPoP/mTLS).

Policy Decision Point der Basisdienste

Der PDP ist die einzige Komponente im System, die Berechtigungsentscheidungen trifft. Er ist konzeptuell vom AS, API Gateway und API (dem jeweiligen Policy Enforcement Point) vollständig getrennt: Der PEP empfängt Client-Anfragen und setzt Entscheidungen durch; der PDP bewertet ausschließlich, ob eine Berechtigung erteilt werden darf bzw. eine Anfrage legitim ist.

Der PDP ist ein **dezentraler, operativ unabhängiger Dienst** innerhalb eines Basisdienstes. Er unterhält keine Online-Verbindung zum zentralen Platform-Directory oder PIP. Stattdessen empfängt er die für seine Auswertungen erforderlichen Policies und Meta-Daten über Auslieferungskanäle des PIP und hält sie lokal vor.

Die Aufgaben des PDP im Überblick:

- Empfang von AuthZEN Evaluation Requests vom AS/API-Gateway
- Auflösung von Client-Attributen und API-Policies in den lokal vorgehaltenen Datenstores
- Auswerten aller aktiven Policies für die angefragte API gegen die Client-Attribute
- Berechnung der Scope-Schnittmenge aus policy-erlaubten und client-beantragten Scopes
- Rückgabe einer AuthZEN Evaluation Response mit Entscheidung, `granted_scopes` und Audit-Informationen

Was der PDP ausdrücklich **nicht** tut:

- Attributwerte außerhalb des Laufzeitkontext vom AS entgegen und deren Richtigkeit überprüfen – alle Attribute stammen direkt aus dem Platform-Directory / PIP
- Token ausstellen – das ist Aufgabe des AS
- Er trifft keine feingranularen Entscheidungen innerhalbeiner API bzw. des Anwendungs-Kontextes – das bleibt beim Resource Server bzw. einem dedizierten PDP der Anwendung

Lokale Datenhaltung, Policy-Updates

Der PDP hält zwei voneinander unterscheidbare lokale Datenspeicher vor, die ausschließlich vom PIP befüllt werden:

1. Policy Store: enthält ausschließlich Berechtigungs-Policies für APIs, für die dieser PDP zuständig ist.
2. Facts Store: enthält ausschließlich Daten und Metadaten über Software & Organisationen, aufgrund der Policy-Kriterien für den PDP als relevant eingestuft wurden. Ein PDP erhält keine Informationen über Clients, die keine seiner APIs betreffen könnten.

Alle Daten werden vom PDP lokal vorgehalten und für die effiziente Regelauswertung zur Laufzeit verwendet. Änderungen an Policies oder Policy-Facts werden über einen sicheren Transportkanal vom PRP/PIP an den PDP ausgeliefert. Die (logische) Trennung der zwei Datenspeicher ermöglicht, eine priorisierte Propagation von Änderungen zum PDP; insbesondere bei sicherheitsrelevanten Änderungen an Attributen (Facts).

Die vom PRP/PIP erhaltenen Policy-/Attribut-Daten sind über eine Versionskennung kryptographische Signaturen geschützt, welche ebenfalls im globalen Transparency-Log veröffentlicht worden sind und so über einen separaten Kommunikationsweg validiert werden. Es werden nur Policy-/Attribut-Daten vom PDP aktiviert, deren Authentizität über das Transparency-Log erfolgreich validiert worden ist.

Durch die lokale Datenhaltung und Caching wird eine effiziente Regelauswertung ermöglicht und Single-Point-of-Failures vermieden (Resilienz): Sollte die zentrale Umgebung (insbesondere der Policy Store) ausfallen/nicht erreichbar sein, können lediglich keine neuen Änderungen an Policy-/Attribut-Daten propagiert werden. Der operative Betrieb bleibt davon unberührt.

Integration in die Infrastruktur

Der Policy-Decision-Point ist die Entscheidungsinstanz für den Authorization Server des Basisdienstes, sowie für ein ggf. vorhandenes API-Gateway (den jeweiligen PEPs).

Die folgende Abbildung illustriert die Interaktion zwischen Authorization Server/API-Gateway und PDP im Rahmen der Berechtigungsprüfung:

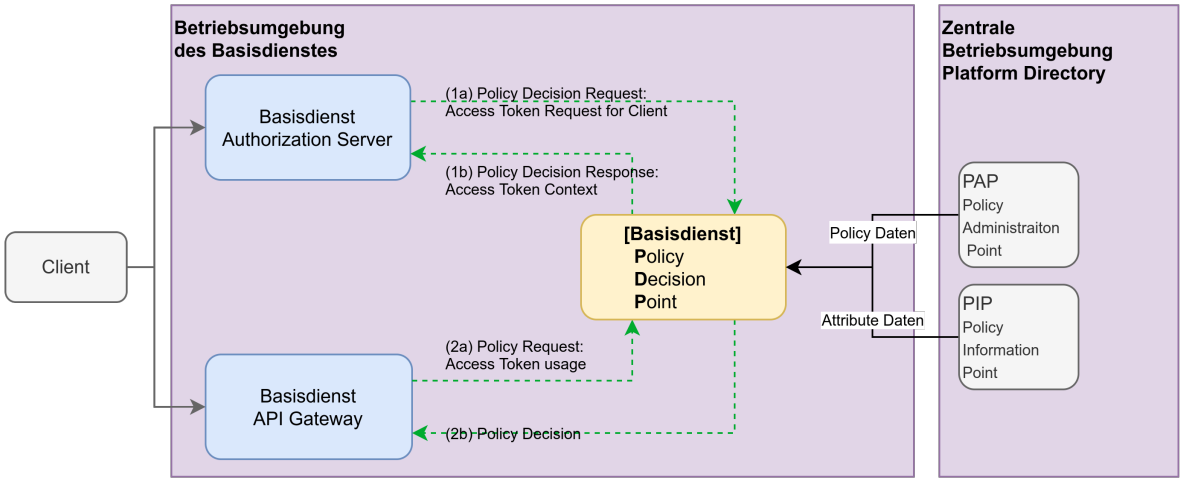


Abbildung 63: Interaktion zwischen Authorization Server/API-Gateway und PDP im Rahmen der Berechtigungsprüfung

Die Abbildung stellt die wesentlichen Informationsflüsse dar:

1 **Basisdienst Authorization Server:** Im Rahmen des Token Exchange überprüft der AS die grundlegende Legitimität der Client-Anfrage (OIDC/FAPI). Anschließend werden dem PDP die Kontextinformationen vorgelegt (1a): der PDP prüft anhand der Policies, ob der AS ein Access Token ausstellen darf und, wenn ja, welche Scope-Berechtigung dieses enthalten darf. Ist das Ergebnis der Policy-Prüfung „Permit", werden die freizugebenden Scopes als Teil Policy-Decision zurückgegeben (1b). Der AS verarbeitet das Ergebnis und stellt - je nach Entscheidung des PDP – ein Access Token mit den jeweiligen Scopes für den Client.

2 **Basisdienst API Gateway** (resp. Resource Server/API): Das API als Resource-Server - oder ein vorgeschaltetes API-Gateway – kann den PDP nutzen um die Legitimität eines bereits ausgestellten Tokens zur Laufzeit gegen die Policy zu überprüfen (2a). Je nach Policy-Ergebnis des PDP –„Permit" oder „Deny" – wird die Client-Anfrage akzeptiert oder abgelehnt. Für legitime Anfragen erfolgt ebenfalls eine Überprüfung der zulässigen Scopes.

Unabhängig vom anfragenden Endpunkt wendet der PDP prinzipiell dieselbe Prüflogik an – ob die Anfrage vom AS oder vom API-Gateway/Resource Server ausgelöst wird.

Interne Verarbeitungsstruktur

Der PDP verarbeitet jeden eingehenden Evaluation Request in den folgenden aufeinanderfolgenden Phasen:

1. Phase – Lokaler Datenlookup: Der PDP schlägt alle benötigten Informationen in seinen lokalen Stores nach; relevante Policies und Facts zum anfragenden Client.
2. Phase – Policy-Auswertung: Für jede aktive Policy wird die Policy-Auswertung durchgeführt. Das Ergebnis ist PERMIT(scopes) oder DENY.
3. Phase – Combining Algorithm: Die Einzelergebnisse werden nach Deny-Overrides-Semantik kombiniert.
4. Phase – Scope-Berechnung: Nur bei Gesamtergebnis PERMIT erfolgt die Berechnung der Schnittmenge aus angefragten und den per Policy freigegeben Scopes. Ist die Schnittmenge leer, ändert sich das Gesamtergebnis zu DENY.

Kommunikationsmodell

PEP-Komponenten (AS/API-Gateway) und PDP kommunizieren über das **AuthZEN-Protokoll** (OpenID Foundation) miteinander. AuthZEN definiert eine einheitliche REST-API zwischen PEP und PDP mit einem Evaluation-Endpunkt für die Auswertung von Berechti-gungsanfragen.

Die Spezifikation ist unter https://openid.net/wg/authzen/specifications/ abrufbar.

Für die Implementierung durch den AS bzw. API-Gateway wird das AuthZEN „Access Evaluation API" verwendet um eine Zugriffsentscheidung beim PDP anzufragen:

Das folgende (nicht-normative) Beispiel illustriert eine Anfrage an den PDP über das AuthZEN-Protokoll:

Evaluation Request (AS → PDP):

```
POST /access/v1/evaluation
Content-Type: application/json

{
  "subject": {
    "type": "software_statement",
    "id":   "urn:platform-directory:ss:a1b2c3d4-e5f6-…",
```

```
    "context": {
      "client_id": "client-id-123….",
      "software_id": "urn:platform-directory:ss:a1b2c3d4-e5f6-f2ba-...",
      "acting_for": {
            "org": {
                  "name": "Freie Musterstadt",
        "id": "https://idp.muk.de/organisations/freie-musterstadt",
            },
            "idp": {
                  "entity_id": "https://idp.muk.de",
            }
      }
    },
  },
  "action": {
    "name": "token_request",
    "context": {
      "requested_scopes": ["lesen", "schreiben"]
    }
  },
  "resource": {
    "type": "api",
    "id":   "urn:api:..."
  },
  "context": {
    "time":   "1970-01-01T23:59-01:00"
  }
}
```

Gemäß der AuthZEN-Spezifikation werden die folgenden Datenfelder verwendet:

- **Subject:** Informationen über den anfragende Client-Software und deren Kontext
- **Action:** Informationen über die auszuführende Aktion und deren Kontext
- **Resource:** Informationen über das Ziel-API auf das der Zugriff erfolgen soll
- **Context:** Kontext-Informationen und Metadaten über die Laufzeitumgebung des Zugriffs

Die Elemente Subject und Resource weisen immer dieselbe Grundstruktur auf: Das Feld „`type`“ trägt den Typ des Subject, „`id`“ die eindeutige Kennung. Beide Attribute sind pflichtangaben im Protokoll. Durch das optionale Unterelement „`context`“ können zusätzliche Daten angereichert werden. Beim Element „`action`“ ist mindestens ein Wert für „`name`“ anzugeben.

Die drei AuthZEN-Kernfelder sind direkt auf das Berechtigungsmodell abgebildet:

AuthZEN-Feld	Beispielwert	Bedeutung
`subject.type`	`software_statement`	Der Typ des zugreifenden Subjects; z. B. "`software_statement`" oder "`user`"
`subject.id`	`urn:platform-directory:ss:a1b2c3d4-e5f6-f2ba-...`	Für API-Berechtigungen ist dies immer der Client bzw. die Kennung des Software-Statement; der PDP löst darüber alle Attribute auf
`subject.context`		Weitere Metadaten des Subjects zur Laufzeit
`resource.type`	`api`	Der Typ der Ressource auf die zugriffen werden soll; bei API-Berechtigungen immer "`api`".
`resource.id`	`urn:platform-directory:api:554372c3-aca0-47c3-...`	Für API-Berechtigungen ist dies die Kennung der angefragten API; der PDP lädt darüber die relevanten Policies und `registered_scopes`
`action.name`	`token_request`	Die auszuführende Aktion im Kontext der API-Abfrage: Ausstellen eines Tokens, bzw. Gültigkeitsprüfung.
`action.context.requested_scopes`	`["lesen", "schreiben"]`	Vom Client beantragte Scopes; Grundlage der Intersection-Berechnung
`context`		*Optional, zusätzliche Metadaten zur Laufzeitumgebung der Anfrage.*

Für die Verwendung von **action.name** im AuthZEN Request werden folgende Werte verwendet:

Anfragende Komponente (PEP)	Wert für „action.name“	Bedeutung der Anfrage
Authorization Server	`client_registration`	Darf ein neuer OAuth-Client an diesem Authorization Server registriert werden? (RFC 7591)
Authorization Server	`token_request`	Darf dem Client ein Token ausgestellt werden? Welchen Inhalt soll es erhalten?

API Gateway	`token_use`	Darf der Client dieses Token mit diesen Werten aktiv für dieses API verwenden?

Für den Kontext der API-Zugriffsprüfung werden keine weiteren Werte für „`action.name`" spezifiziert.

Ungültige Anfragen werden gemäß der Default-Einstellung des PDP mit „Deny" beantwortet.

Laufzeitattribute und Subjektkontext

Clients werden im Rahmen der Berechtigungsprüfung primär über ihr Software Statement repräsentiert; in manchen Fällen liegen die konkreten Identitätsinformationen jedoch erst zur Laufzeit beim AS vor (z. B. beim Authorization Code Flow, wenn ein *Nutzer* einer Org einen Antrag stellt bzw. einen Onlinedienst nutzt).

Typische (nicht-normative) Beispiele solcher Laufzeitattribute sind:

- `org.id` – die organisatorische Kennung der anfragenden Stelle,
- `idp.entity_id` – der Identitätsprovider, über den der Client authentisiert wurde,
- `org.funktionskennzeichen` – ein fachliches Kennzeichen zur Beschreibung der Funktion oder Zuständigkeit der Organisation.

Der AS übermittelt diese Attribute über `subject.context` an den PDP. Der PDP wertet die Kontextattribute zusätzlich zu den intern vorliegenden, statischen Attributen aus (insbes. der auf Software Statements basierenden). Policies können damit Kriterien beider Attributquellen kontextualisieren.

Die Verwendung von `subject.context` ist durch den zentralen Attributkatalog normiert: Ausschließlich dort definierte Attribute können in Policies referenziert und zur Laufzeit ausgewertet werden. Dies stellt sicher, dass auch dynamische Laufzeitattribute konsistent, kanonisiert und plattformweit einheitlich verwendet werden.

Das folgende (nicht-normative) Beispiel illustriert eine AuthZEN-Anfrage an den PDP im Kontext einer Delegation („Authorization Code Flow“):

Evaluation Request (AS → PDP):

POST /access/v1/evaluation
Content-Type: application/json

```
{
  "subject": {
    "type": "user",
    "id":   "erika.mustermann@my-identity.example",
    "context": {
      "client_id": "client-id-123….",
      "software_id": "urn:platform-directory:ss:a1b2c3d4-e5f6-f2ba-...",
      "acting_for": {
            "org": {
                 "name": "Freie Musterstadt",
        "id": "https://idp.muk.de/organisations/freie-musterstadt",
            },
            "idp": {
                 "entity_id": "https://idp.muk.de",
            }
      }
    },
  },
  "action": { ... },
  "resource": { ... },
  "context": { ... },
}
```

Über die Attribute "`subject.type`" "`subject.id`" wird der Nutzertyp und dessen Kennung referenziert, die Attribute in "`subject.context`" repräsentieren die Attribute zum Laufzeitkontext.

Beispiel - Antworten des PDP:

Evaluation Response (PDP → AS) bei PERMIT:

HTTP/1.1 200 OK
Content-Type: application/json

```
{
  "decision": true,
  "context": {
    "granted_scopes":     ["lesen"],
    "matched_policy_ids": ["policy-example-permit-123"],
    "reason": "Berechtigung durch PERMIT-Policy"
  }
}
```

"`decision: true`" entspricht PERMIT aus dem Combining Algorithm. Über das (optionale) Element „`context`“ werden die von der Policy ermittelten Scopes mitgegeben, welche der AS für die Scope-Berechtigung in das Access-Token kodieren kann.

Weiterhin werden Debug- und Fehlerinformationen über Attribute im „`context`“-Element kommuniziert.

Evaluation Response (PDP → AS) bei DENY:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "decision": false,
  "context": {
    "granted_scopes": [],
    "matched_policy_ids": ["policy-example-deny-123"],
    "reason": "Explizite Ablehnung durch DENY-Policy"
  }
}
```

„`decision: false`" umfasst sowohl explizites DENY als auch Implicit Deny und den Fall einer leeren Scope-Intersection.

Offene Punkte

[Lesehinweis: Zum aktuellen Stand der Konzeption bedürfen die folgende Aspekte weiter Konkretisierung. Dies erfolgt in den kommenden Iterationen des Konzepts]

- Verbindliches Mapping zwischen Katalog- und Laufzeitattributen auf die jeweiligen Felder der Policy-Struktur und der AuthZEN-Anfrage
- Definition des Attributmappings zur kontextuellen Unterscheidung der relevanten Anwendungsfälle:
  - Client-Credentials: Maschine-zu-Maschine-Kommunikation ohne aktiven Nutzerkontext – der Client handelt eigenständig und im Auftrag einer Organisation
  - Authorization Code Flow:
    - Eine natürliche Person nutzt einen Online-Dienst direkt in eigenem Namen
    - Ein Service-Client wird durch eine Organisation betrieben und handelt im Kontext einer Behörde oder Organisation

5.2 Übergreifendes Monitoring und Risikobewertung

[Lesehinweis: Zum aktuellen Stand der Konzeption ist die Risikobewertung und das Monitoring noch nicht in das Berechtigungskonzept integriert. Das erfolgt erst in kommenden Iterationen des Konzepts]

Die föderale API-Autorisierungsinfrastruktur verteilt sicherheitsrelevante Entscheidungen und Ereignisse strukturbedingt über eine Vielzahl dezentral betriebener Komponenten: Authorization Server, API-Gateways und Policy Decision Points liegen in der Betriebsverantwortung der jeweiligen Basisdienste. Diese Dezentralität ist architektonisch gewollt (Subsidiaritätsprinzip, Betriebsautonomie der Basisdienste), stellt aber gleichzeitig eine Herausforderung für die übergreifende Sicherheitsüberwachung dar: Isoliert betrachtet ist ein einzelnes Ereignis wie etwa ein häufig abgelehnter Token-Request eines API-Consumer-Clients möglicherweise unauffällig. Im Kontext aller Basisdienste kann dasselbe Ereignismuster ein Indikator für eine Kompromittierung oder einen Missbrauchsversuch sein.

Das vorliegende Kapitel beschreibt, wie diese übergreifende Monitoring- und Risikobewertungsfunktion durch den Einsatz des **Shared Signals Framework (SSF)** der OpenID Foundation realisiert wird. Es steht in engem Bezug zur Fähigkeitsgruppe Logging und Monitoring (siehe Kapitel 3.3), die die strategischen Fähigkeiten Ereignisprotokollierung, Anomalieerkennung, Betriebsüberwachung und Berichterstattung umfasst, und legt dar, wie diese Fähigkeiten infrastrukturell umgesetzt und Basisdienst-übergreifend koordiniert werden.

5.2.1 Shared Signals Framework als föderale Signalschicht

Das **Shared Signals Framework (SSF)** ist ein Standard der OpenID Foundation, der einen standardisierten, sicherheitsgesicherten Kanal für den Austausch sicherheitsrelevanter Ereignissignale zwischen Systemkomponenten definiert. Die drei Kernspezifikationen SSF 1.0, das Continuous Access Evaluation Profile (CAEP) und das Risk Incident Sharing and Coordination Profile (RISC) wurden im September 2025 als finale Spezifikationen verabschiedet.

Das SSF-Modell unterscheidet zwischen **Transmittern** (Ereignisquellen) und **Receivern** (Ereignisempfängern), die über typisierte **Streams** verbunden sind. Ereignisse werden als **Security Event Tokens (SETs)** gemäß RFC 8417 übertragen, einem signierten JWT mit standardisierten Subject-Identifiern. Die Streams unterstützen Push-Delivery (RFC 9835) und Poll-Delivery (RFC 8936), was unterschiedliche Betriebsmodelle der Basisdienste berücksichtigt.

Im Kontext der föderalen API-Autorisierungsinfrastruktur wird das SSF als **föderale Signalschicht** eingesetzt, die:

- sicherheitsrelevante Ereignisse der dezentralen Basisdienst-Komponenten sowie der zentralen Infrastrukturkomponenten bündelt und korreliert,
- Risikosignale normiert und für nachgelagerte Komponenten maschinell verarbeitbar macht,
- die automatische Reaktion auf Sicherheitsereignisse in den dezentralen Policy-Infrastrukturen der Basisdienste (z. B. unmittelbare Verweigerung des API-Zugriffs auf Basis eines erhöhten Risikowerts einer Software) sowie in der zentralen Policy-Infrastruktur und im FöPD auslösen kann,
- und die Integration mit externen Sicherheitssystemen und Auditplattformen ermöglicht.

5.2.2 Grundarchitektur der SSF-basierten Überwachungsinfrastruktur

Die SSF-Überwachungsinfrastruktur gliedert sich in drei Schichten.

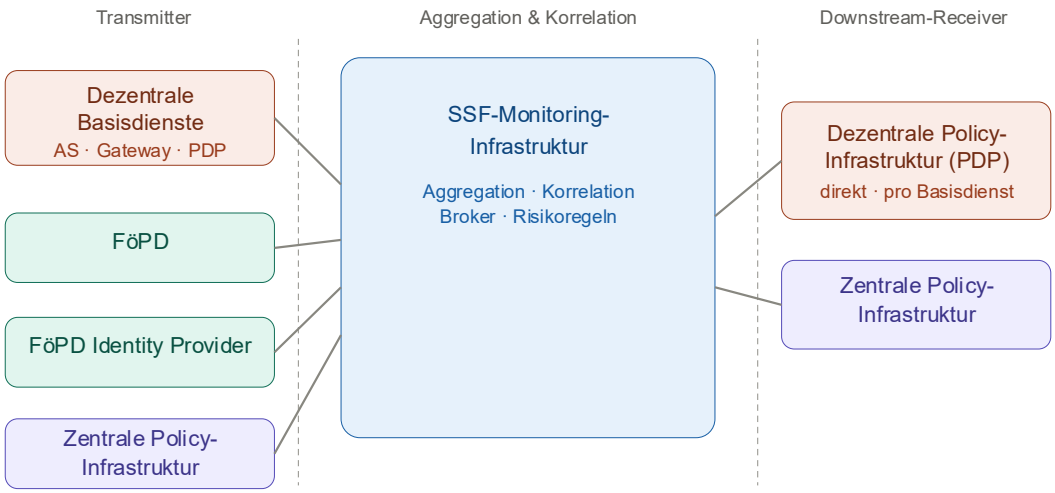


Abbildung 64: Schichten der SSF-Architektur

Schicht 1 – Ereignisquellen (Transmitter)

Alle zentralen und dezentralen Kerninfrastrukturkomponenten agieren als SSF-Transmitter. Die dezentralen Basisdienst-Komponenten übermitteln ihre Ereignisse über einen jeweils lokalen SSF-Transmitter-Adapter (vgl. Kapitel 4.2). Relevante Ereignisquellen sind bspw.:

- **[Basisdienst] Authorization Server:** Token-Ausstellungen, abgelehnte Token-Requests, Token-Exchange-Vorgänge, Fehlversuche

- **[Basisdienst] API-Gateway:** abgelehnte API-Anfragen (ungültiges Token, DPoP-Fehler, Scope-Mismatch), Anomaliemuster im Request-Volumen
- **[Basisdienst] Dezentrale Policy-Infrastruktur:** Policy-DENY-Entscheidungen, Replikationsfehler, lokale Regelverstöße
- **FöPD:** administrative Ereignisse wie Registrierungsänderungen, Client- und Organisationssperrungen, Entzug von API-Zugriffsberechtigungen, Policy-Anpassungen
- **FöPD Identity Provider:** Authentifizierungsereignisse von FöPD-Nutzern, fehlgeschlagene Anmeldeversuche, Änderungen an Organisationsattributen und Nutzeraccounts, Statusänderungen von IdP-Föderierungen sowie Ereignisse im Kontext der Attributaggregation aus externen Quellen (BundID, MUK, externe IdPs)
- **Zentrale Policy-Infrastruktur:** Policy-Änderungssignale, Replikationsstatus, Integritätsereignisse

Die zentralen Infrastrukturkomponenten schreiben ihre sicherheits- und administrativ relevanten Ereignisse **parallel** in den Transparency Log und über den jeweiligen SSF-Transmitter-Adapter in die SSF-Monitoring-Infrastruktur. Diese Parallelität ist architektonisch bewusst gewählt: Der Transparency Log bildet den kryptographisch verifizierbaren, manipulationsresistenten Audit-Pfad für unabhängige Prüfinstanzen; die SSF-Monitoring-Infrastruktur den operativen Reaktionskanal für Echtzeit-Risikobewertung und automatische Gegenmaßnahmen. Die Verfügbarkeit und Integrität des Audit-Pfads sind damit strukturell unabhängig vom Betrieb der operativen Monitoring-Infrastruktur. Die konzeptionelle Abgrenzung beider Infrastrukturkomponenten wird in Kapitel 5.3 vertieft.

Schicht 2 – Aggregation und Korrelation (zentraler Receiver)

Die **zentrale SSF-Monitoring-Infrastruktur** (vgl. Kapitel 4.2) empfängt alle Streams der dezentralen Basisdienst-Adapter sowie der zentralen Infrastrukturkomponenten. Sie korreliert Ereignisse über Subjekte (API-Consumer-Clients, Organisationen) und Basisdienste hinweg und bewertet diese anhand konfigurierbarer Risikoregeln. Bei Überschreitung definierter Schwellwerte werden typisierte Risikosignale an die nachgelagerten Receiver-Komponenten weitergeleitet.

Schicht 3 – Reaktion (Downstream-Receiver)

Auf Basis der aggregierten Risikosignale werden parallel automatische Reaktionen ausgelöst:

- **Dezentrale Policy-Infrastruktur der Basisdienste:** Der jeweils lokale PDP empfängt das Risikosignal direkt als SSF-Receiver und aktualisiert seine Entscheidungsgrundlage unmittelbar, etwa durch Aktivierung einer DENY-Entscheidung auf Basis eines erhöhten Risikowerts einer Software oder einer Sperrung. Dieser direkte Signalkanal ergänzt den regulären PIP-Auslieferungsweg und ermöglicht eine zeitnahe Reaktion unabhängig vom Policy-Replikationszyklus. Die Erweiterung des im Berechtigungskonzept beschriebenen Auslieferungsmodells um diesen dynamischen SSF-Kanal ist als offener Punkt dokumentiert (vgl. Kapitel 5.2.4).
- Zentrale Policy-Infrastruktur: Aktivierung oder Änderung einer übergreifenden DENY-Policy (z. B. Setzen von `software.locked = true` als Policy Condition), die anschließend über den regulären PIP-Kanal in die dezentralen Policy-Infrastrukturen aller betroffenen Basisdienste repliziert wird.
- **FöPD:** Auslösung administrativer Workflows, z. B. Benachrichtigung der Betriebsverantwortlichen Stelle oder Einleitung eines Prüfprozesses.
- **Externe Sicherheitssysteme:** Empfang der aggregierten Signale über einen dedizierten SSF-Stream für weitergehende Sicherheitsanalysen und Auditierung.

5.2.3 Notwendigkeit von Eventprofilierung und geeigneter Subject Identifikatoren

Das SSF definiert mit CAEP und RISC zwei Standardprofile. CAEP adressiert primär Statusänderungen aktiver Sessions (z. B. Token-Revokation, Credential-Änderungen), RISC den Austausch von Risikoereignissen rund um Account-Kompromittierungen. Beide Profile sind auf natürliche Personen und deren Sessions ausgerichtet.

Im Kontext der föderalen API-Autorisierungsinfrastruktur decken diese Standardprofile den Bedarf nur partiell ab: Relevante Ereignisse betreffen hier primär maschinelle Subjekte (registrierte Software, API-Consumer-Clients) und organisatorische Subjekte (Betreiber-Organisationen oder Basisdienstnutzer) sowie infrastrukturspezifische Vorgänge wie administrative Sperrungen, Policy-Änderungen oder basisdienstübergreifend erkannte Anomaliemuster. Für diese Ereignisklassen sind **föderale API-spezifische SSF-Eventprofile** zu spezifizieren, die über CAEP und RISC hinausgehen (siehe Kapitel 5.2.4).

Subject-Typen

Das SSF erlaubt neben natürlichen Personen ausdrücklich auch maschinelle und organisatorische Subjekte als Subject Principals. Für die Identifikation von Subjects in der föderalen Signalschicht gilt folgendes Prinzip: Die `client_id` eines API-Consumer-Clients wird vom jeweiligen Authorization Server eines Basisdienstes bei der Registrierung lokal vergeben und ist damit nur im Kontext des jeweiligen Basisdienstes eindeutig. Für basisdienst-übergreifende SSF-Signale ist sie daher nicht als Subject-Identifier geeignet.

Stattdessen werden Clients infrastrukturweit über ihre **Software ID des FöPD** identifiziert, die als globaler, plattformweiter Identifier für jede registrierte Software eindeutig ist und von allen Infrastrukturkomponenten aufgelöst werden kann. Folgende Subject-Identifier-Typen kommen zum Einsatz:

- `uri` mit FöPD-Software-ID zur infrastrukturweiten Identifikation von API-Consumer-Clients und registrierten Softwareinstanzen
- `uri` mit FöPD-Organisations-ID oder einem anderen geeigneten Identifier zur Identifikation von Organisationen
- `did` oder `iss_sub` als optionale zukünftige Erweiterungen in Richtung dezentraler Identitäten

5.2.4 Offene Themen

Die Umsetzung der SSF-basierten Monitoring-Infrastruktur erfordert die Bearbeitung mehrerer offener Punkte, die zusammen mit den weiteren offenen Fragen und Handlungsbedarfen des Vorhabens in Kapitel 7.5 gebündelt sind.

5.3 Revisionssichere Protokollierung und Auditierung

Kapitel 5.2 beschreibt, wie die föderale API-Autorisierungsinfrastruktur sicherheitsrelevante Ereignisse über alle Basisdienste hinweg in Echtzeit aggregiert, bewertet und auf erkannte Risiken operativ reagiert. Der Schwerpunkt liegt auf der Reaktionsfähigkeit: Signale fließen, Entscheidungen werden angepasst, Sperren werden gesetzt.

Das vorliegende Kapitel richtet den Blick auf eine komplementäre, aber konzeptionell eigenständige Anforderung: die **Beweissicherung**. Operative Reaktionsfähigkeit und revisionssichere Beweissicherung sind zwei unterschiedliche Qualitäten, die unterschiedliche infrastrukturelle Antworten erfordern und nicht ineinander aufgehen dürfen.

Die zentrale Unterscheidung ist:

Die **SSF-Monitoring-Infrastruktur** (Kapitel 5.2) beantwortet die Frage: *Was muss jetzt operativ reagieren?* Sie ist ein Streaming-System, welches konfigurierbar, konsumierbar und auf Echtzeit ausgelegt ist. Ihre Signale sind aggregierte, korrelierte Urteile über Ereignismuster. Sie ist kein geeignetes Beweismittel, weil Konfigurationsänderungen, Ausfälle und Verarbeitungslogik die Vollständigkeit und Integrität der weitergeleiteten Signale beeinflussen.

Die **Transparency Log Infrastruktur** (vgl. Kapitel 4.2) beantwortet die Frage: *Was ist in der zentralen Infrastruktur nachweislich passiert?* Sie ist ein append-only, kryptographisch verifizierbares System, das unveränderlich, vollständig, unabhängig vom Vertrauen in den Betreiber prüfbar. Ihr Inhalt sind Rohereignisse der zentralen Infrastrukturkomponenten, keine abgeleiteten Urteile.

Diese Trennung ist für den föderalen Kontext besonders bedeutsam: An der föderalen API-Autorisierungsinfrastruktur sind viele Betreiber mit unterschiedlichem Vertrauensniveau beteiligt. Ein Audit-System, das auf dem Vertrauen in den Betreiber des Logs basiert, genügt den Anforderungen einer föderalen, behördenübergreifenden Infrastruktur nicht.

5.3.1 Beweissicherung der zentralen Infrastruktur

Der Transparency Log erfasst ausschließlich Ereignisse der **zentralen Infrastrukturkomponenten**, nicht aber der dezentralen Basisdienst-Komponenten.

Diese Scoping-Entscheidung folgt der Vertrauensarchitektur der Gesamtinfrastruktur: Die dezentralen Basisdienste sind Konsumenten der zentralen Infrastruktur. Sie vertrauen auf die Integrität der Daten, die sie vom FöPD und der zentralen Policy-Infrastruktur erhalten, wie insbesondere Software Statement Assertions, Policies und Attributinformationen. Eine Manipulation dieser zentralen Daten würde von den Basisdiensten unbemerkt bleiben, da diese keine unabhängige Möglichkeit haben, die Integrität der zentralen Datenbasis zu verifizieren.

Der Transparency Log sichert genau diesen Vertrauensanker ab: Er macht nachweisbar, was die zentrale Infrastruktur tatsächlich ausgestellt, entschieden und verändert hat. Er stellt damit sicher, dass Manipulationen oder unbemerkte Eingriffe in die zentrale Infrastruktur nicht dauerhaft verborgen bleiben können.

Ob und in welchem Umfang Basisdienste eine eigene lokale Protokollierung betreiben, liegt in ihrer jeweiligen Betriebsverantwortung und ist nicht Gegenstand dieses Konzepts.

5.3.2 Eigenschaften des Transparency Logs

Der Transparency Log der föderalen API-Autorisierungsinfrastruktur basiert auf einem kryptographisch verifizierbaren Merkle-Tree. Die wesentlichen Eigenschaften sind:

- **Append-only:** Einträge können dem Log nur hinzugefügt, aber weder geändert noch gelöscht werden. Dies gilt auch gegenüber dem Betreiber des Logs selbst.
- **Kryptographische Verifikation:** Jeder Zustand des Logs wird durch einen signierten Checkpoint repräsentiert. Jeder Eintrag kann durch einen Inclusion Proof nachweislich als Teil des Logs belegt werden. Die Konsistenz zwischen zwei Zuständen des Logs lässt sich durch einen Consistency Proof verifizieren, ohne Vertrauen in den Betreiber des Logs haben zu müssen.
- **Unabhängige Auditierbarkeit:** Externe Prüfinstanzen können das Log vollständig lesen, seine Integrität eigenständig verifizieren und Einträge gegen Inclusion Proofs prüfen. Die Lesbarkeit des Logs ist strukturell von der schreibenden Infrastruktur getrennt.
- **Kein Widerrufsrecht:** Da Einträge nicht gelöscht werden können, sind auch irrtümlich oder missbräuchlich vorgenommene administrative Akte dauerhaft nachweisbar. Das ist eine bewusste Designentscheidung zugunsten der Nachvollziehbarkeit.

5.3.3 Protokollierte Ereignisklassen

Der Transparency Log erfasst ausschließlich **Rohereignisse der zentralen Infrastrukturkomponenten** und keine aggregierten oder abgeleiteten Signale der SSF-Monitoring-Infrastruktur sowie keine Ereignisse der dezentralen Basisdienst-Komponenten.

Administrative Ereignisse des FöPD:

- Ausstellung und Revokation von Software Statement Assertions
- Client-Sperrungen und -Entsperrungen
- Organisations-Sperrungen
- Entzug und Wiederherstellung von API-Zugriffsberechtigungen
- Registrierungsänderungen an Software-Einträgen und Plattformangeboten

Berechtigungshistorie der zentralen Policy-Infrastruktur:

- Erstellung, Änderung und Löschung von Policies
- Replikationsstatus und ausgelieferte Policy-Versionen je PDP-Instanz

Ereignisse des FöPD Identity Providers:

- Änderungen an Organisationsattributen und deren LoA-Einstufung
- Föderierungsänderungen (Hinzufügen, Ändern, Entfernen externer IdP-Verbindungen)

Ausdrücklich nicht protokolliert werden: Ereignisse dezentraler Basisdienst-Komponenten (Authorization Server, API-Gateway, dezentrale Policy-Infrastruktur) sowie aggregierte Signale und abgeleitete Urteile der SSF-Monitoring-Infrastruktur (z. B. Risk Scores, erkannte Anomaliemuster).

5.3.4 Parallele Befüllung und Unabhängigkeit der Infrastrukturpfade

Die zentralen Infrastrukturkomponenten schreiben ihre Ereignisse parallel in den Transparency Log und über die SSF-Monitoring-Infrastruktur in den operativen Reaktionskanal. Beide Pfade sind betrieblich entkoppelt: Ein Ausfall der SSF-Monitoring-Infrastruktur darf keine Auswirkung auf die Integrität oder Vollständigkeit des Transparency Logs haben, und umgekehrt. Diese strukturelle Entkopplung ist Voraussetzung dafür, dass der Transparency Log als unabhängiges Beweismittel anerkannt werden kann.

Die folgende Gegenüberstellung fasst die konzeptionelle Unterscheidung zusammen:

Tabelle 38: Unterschied Transparency Log und SSF-Monitoring

	Transparency-Log	SSF-Monitoring-Infrastruktur
Zweck	Beweissicherung, Auditierung	Operative Reaktion, Risikobewertung
Schreibende Komponenten	Ausschließlich zentrale Infrastruktur	Zentrale und dezentrale Infrastruktur
Inhalt	Rohereignisse	Aggregierte und korrelierte Signale
Schreibmodell	Append-only, unveränderlich	Streaming, konfigurierbar
Integrität	Kryptographisch verifikabel (Merkle-Tree)	Operational, nicht kryptographisch garantiert
Leserkreis	Unabhängige Prüfinstanzen, Auditoren	Downstream-Receiver: PDPs, Policy-Infrastruktur, FöPD, SIEM
Vertrauen	Vertrauensunabhängig verifizierbar	Setzt Vertrauen in Betreiber und Konfiguration voraus

Latenz	Nicht echtzeit-kritisch	Echtzeit

5.3.5 Zugang für externe Auditoren

Der Transparency Log ist über eine lesende API für externe Prüfinstanzen zugänglich (vgl. Drittsysteme von Auditoren in Kapitel 4.2). Prüfinstanzen können:

- das gesamte Log lesen und lokal verifizieren,
- Inclusion Proofs für einzelne Einträge anfordern,
- Consistency Proofs zwischen zwei Log-Zuständen prüfen,
- und die Signatur des Checkpoints gegen den öffentlichen Schlüssel des Log-Betreibers verifizieren.

Dieser Zugang ist lesend und erfordert keine privilegierten Rechte.

5.3.6 Lösungsauswahl

Das vorliegende Konzept enthält an dieser Stelle eine Vorfestlegung auf eine konkrete Implementierungslösung mit der Implementierungentscheidung die Transparency Log Infrastruktur auf Basis von **Tessera** (transparency-dev/tessera).

Sie ergibt sich aus der besonderen Natur der Anforderung: Die technischen Eigenschaften eines Transparency Logs wie kryptographische Verifikation, append-only Struktur und vertrauensunabhängige Auditierbarkeit sind keine optionalen Qualitätsmerkmale, sondern konstitutiv für den Zweck der Komponente. Ein System, das diese Eigenschaften nicht mitbringt, ist kein Transparency Log, sondern ein konventionelles Audit-Log, das die hier formulierten Sicherheitsziele nicht erfüllt.

Warum keine Eigenentwicklung?

Eine Eigenentwicklung eines kryptographisch verifizierbaren Merkle-Tree-basierten Logs scheidet aus. Die kryptographischen Protokolle für Inclusion Proofs, Consistency Proofs und signierte Checkpoints sind zwar spezifiziert, aber ihre korrekte Implementierung erfordert spezialisiertes Fachwissen und einen erheblichen Validierungsaufwand. Fehler in der kryptographischen Kernimplementierung würden genau die Eigenschaft untergraben, die den Transparency Log vom konventionellen Audit-Log unterscheidet. Im öffentlichen Sektor wäre eine nicht ausreichend erprobte Eigenentwicklung in diesem Bereich schwer vertretbar.

Warum keine konventionelle Audit-Log-Lösung?

Konventionelle Audit-Log-Lösungen, ob als Komponente eines Sicherheitsmanagementsystems, als Datenbankaudit-Feature oder als strukturiertes Logging-Framework, erfüllen die hier gestellten Anforderungen strukturell nicht. Sie bieten keine kryptographisch verifizierbaren Inclusion Proofs, keine Consistency Proofs und keine vertrauensunabhängige Auditierbarkeit. Ihre Integrität setzt Vertrauen in den Betreiber und die Betriebsumgebung voraus, genau das, was im föderalen Kontext mit mehreren beteiligten Stellen unterschiedlichen Vertrauensniveaus nicht vorausgesetzt werden kann.

Warum Tessera und nicht Trillian v1?

Trillian v1 (google/trillian) ist der direkte Vorläufer von Tessera und wird in Produktionsumgebungen eingesetzt. Tessera ist jedoch der designierte Nachfolger: Es baut auf den Erfahrungen eines Jahrzehnts mit Trillian v1 auf, vereinfacht Betrieb und Deployment erheblich, und setzt konsequent auf die tlog-tiles API, einen offenen, von mehreren Ökosystemen gemeinsam genutzten Standard, der eine erheblich bessere Cachebarkeit und Leseskalierung ermöglicht. Neue Ökosysteme, die einen Transparency Log aufbauen, sollten Tessera bevorzugen; eine Entscheidung für Trillian v1 würde eine absehbare Migrationslast erzeugen.

Warum keine alternativen Open-Source-Implementierungen?

Im relevanten Lösungsraum Open-Source-Implementierungen eines kryptographisch verifizierbaren, append-only Merkle-Tree-basierten Logs mit offener Log-Struktur, existiert nach aktuellem Kenntnisstand keine gleichwertige Alternative zu Tessera:

- **Rekor** (sigstore/rekor) ist eine Transparency-Log-Implementierung im Kontext von Sigstore und auf den Software-Supply-Chain-Use-Case ausgerichtet. Die Log-Personality ist auf Artefakte der Code-Signierung zugeschnitten; eine Anpassung auf administrative Infrastrukturereignisse wäre möglich, entspräche aber nicht dem primären Einsatzzweck des Projekts.
- **Sunlight, Azul, Itko** und vergleichbare Implementierungen sind ausschließlich auf Certificate Transparency ausgerichtet. Sie sind keine allgemein verwendbaren Log-Frameworks.
- Kommerzielle Lösungen mit vergleichbaren kryptographischen Eigenschaften scheiden im Kontext der öffentlichen Verwaltung und des Open-Source-Gebots grundsätzlich aus.

Tessera ist damit die einzige bekannte Open-Source-Lösung, die einen allgemein verwendbaren, kryptographisch verifizierbaren Transparency Log mit beliebiger Log-Personality bereitstellt, aktiv weiterentwickelt wird und einen klaren Produktionspfad aufweist.

Vorbehalt und Konsequenz

Die Vorfestlegung auf Tessera erfolgt unter dem in Kapitel 5.3.7 dokumentierten Vorbehalt des Erprobungsbedarfs. Tessera ist für den hier beschriebenen Use Case technisch geeignet, aber die Produktionserfahrung konzentriert sich bisher auf andere Ökosysteme. Die Vorfestlegung bedeutet daher nicht, dass Tessera ohne weiteren Proof-of-Concept unmittelbar produktiv eingesetzt werden soll.

5.3.7 Offene Themen

Die produktive Umsetzung der Transparency-Log-Infrastruktur erfordert die Bearbeitung mehrerer offener Punkte, die zusammen mit den weiteren offenen Fragen und Handlungsbedarfen des Vorhabens in Kapitel 7.6 gebündelt sind.

5.4 Betriebsfragen

5.4.1 Umgang mit zentralen Systemen

Die zentrale Infrastruktur der föderalen API-Autorisierungsinfrastruktur wird unter der operativen Steuerung der Plattformverantwortlichen Stelle betrieben. Die Plattformverantwortliche Stelle ist dabei einer föderalen Steuerungsebene rechenschaftspflichtig, die als Auftraggeber und strategische Steuerungsinstanz die übergeordneten Ziele, Rahmenbedingungen und Entwicklungsprioritäten der Plattform vorgibt.

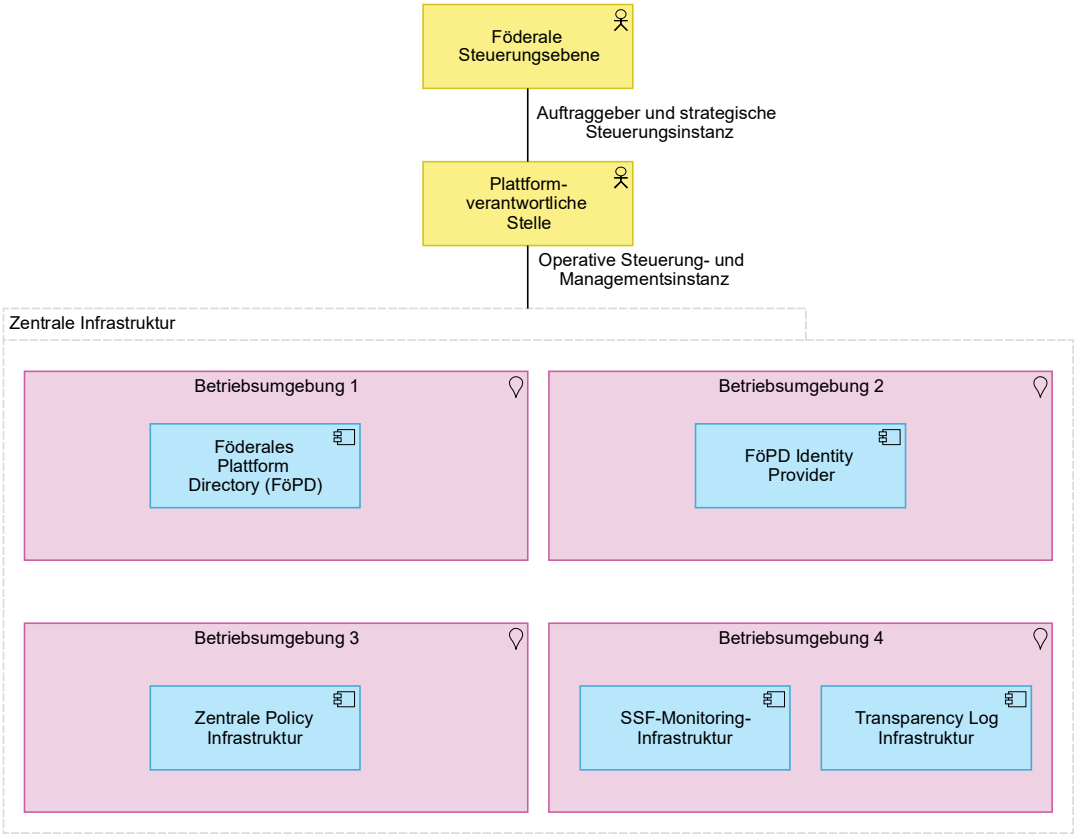


Abbildung 65: Verantwortlichkeiten und Strukturierung der Betriebsumgebungen der zentralen Infrastruktur

Die zentralen Systeme der Kerninfrastruktur werden in vier eigenständige Betriebsumgebungen gegliedert, die jeweils einen klar abgegrenzten Verantwortungsbereich adressieren:

Tabelle 39: Betriebsumgebungen der Kerninfrastruktur

Betriebsumgebung	System(e)	Verantwortungsbereich
Betriebsumgebung 1	Föderales Plattform Directory (FöPD)	Verwaltung und Koordination: Registrierung von Organisationen, Software und API-Clients, Katalogverwaltung, Workflow-Steuerung und Ausstellung von Vertrauensnachweisen
Betriebsumgebung 2	FöPD Identity Provider	Identität und Authentifizierung: Verwaltung von Nutzeraccounts und Organisationsattributen, Authentifizierung von FöPD-Nutzern und Aggregation von Identitätsdaten aus externen Quellen

Betriebsumgebung 3	Zentrale Policy Infrastruktur	Autorisierungsregeln und Entscheidungsgrundlage: Verwaltung und Bereitstellung der kanonischen Berechtigungsregeln, des Attribute Stores und der Datengrundlage für die dezentrale Replikation
Betriebsumgebung 4	SSF-Monitoring-Infrastruktur und Transparency Log Infrastruktur	Beobachtung und Nachweisführung: Operative Echtzeit-Risikobewertung sowie manipulationssichere Protokollierung sicherheitskritischer Ereignisse als revisionssicherer Prüfpfad

5.4.1.1 Begründung der Clusterung

Die Aufteilung in vier separate Betriebsumgebungen folgt drei architektonischen Leitgedanken:

- *Gegenseitige Absicherung durch strukturelle Unabhängigkeit.* Die vier Betriebsumgebungen bilden ein System wechselseitiger Kontrolle. Im Normalbetrieb melden alle Systeme ihre sicherheits- und administrativ relevanten Ereignisse sowohl an die Transparency Log Infrastruktur als auch an die SSF-Monitoring-Infrastruktur. Durch die Trennung in eigenständige Betriebsumgebungen mit jeweils eigenen Betriebsverantwortlichkeiten entsteht eine Architektur, in der jedes System die Handlungen der anderen Systeme über die dokumentierten Datenmeldungen beobachten und verifizieren kann. Ein Angreifer, der eine einzelne Betriebsumgebung kompromittiert, erzeugt unweigerlich Inkonsistenzen in den Datenmeldungen der übrigen drei Umgebungen, sei es durch fehlende Ereignisse im Transparency Log, durch abweichende Risikosignale in der SSF-Monitoring-Infrastruktur oder durch nicht plausible Zustandsänderungen in den Policy- oder Identitätsdaten. Erst die gleichzeitige und unbemerkte Kontrolle aller vier Betriebsumgebungen würde es einem Angreifer ermöglichen, einen Eingriff vollständig zu verschleiern. Die bewusste Zuordnung der Beobachtungs- und Nachweisführungssysteme in eine eigene, von den beobachteten Systemen getrennte Betriebsumgebung stellt dabei sicher, dass die Überwachungsinfrastruktur strukturell nicht vom Überwachungsgegenstand abhängig ist.
- *Unabhängige Entwicklungsfähigkeit.* Die Clusterung orientiert sich an vier orthogonalen Verantwortungsbereichen (Verwaltung, Identität, Autorisierung und Beobachtung), die jeweils eigenständige fachliche und technische Domänen darstellen. Diese Trennung ermöglicht es, jede Betriebsumgebung unabhängig weiterzuentwickeln, zu skalieren und gegebenenfalls, um Aufgaben zu erweitern, die über den unmittelbaren Scope der API-

Autorisierungsinfrastruktur hinausgehen. So könnte etwa der FöPD Identity Provider perspektivisch als zentraler Identitätsdienst für weitere Plattformfunktionen dienen oder die zentrale Policy Infrastruktur für zusätzliche Berechtigungsdomänen genutzt werden, ohne dass dies Auswirkungen auf die anderen Betriebsumgebungen hätten.

- *Kontraktbasierte Zusammenarbeit und Austauschbarkeit.* Die Architektur erzwingt eine präzise Definition der Schnittstellen und Datenkontrakte zwischen den vier Betriebsumgebungen. Die Systeme kommunizieren ausschließlich über dokumentierte und standardisierte Schnittstellen, was die Kopplung auf die Kontraktebene beschränkt. Dies schafft die Voraussetzung, einzelne Systeme bei veränderten Rahmenbedingungen durch alternative Lösungen zu ersetzen, ohne die Gesamtarchitektur grundlegend anpassen zu müssen. Ein Beispiel hierfür wäre eine künftige Ablösung des FöPD durch eine Wallet-basierte Vertrauensinfrastruktur, sollte die Entwicklung der EU Digital Identity Wallet dies ermöglichen.

5.4.2 Umgang mit dezentralen Systemen

Dezentrale Systeme der Kerninfrastruktur übernehmen zentrale Aufgaben der föderalen API-Autorisierungsinfrastruktur, liegen aber in der Betriebsverantwortung der jeweiligen Basisdienst-Betriebsorganisationen (siehe Kapitel 4.2). Für die Bereitstellung dieser Systeme unterscheidet die Zielarchitektur zwei Modelle: die **verbindliche Bereitstellung** einer einheitlichen Lösung und die **optionale Bereitstellung** einer Referenzlösung mit der Möglichkeit, basisdienst-eigene Lösungen einzusetzen.

Die folgende Tabelle gibt einen Überblick über die Einordnung der dezentralen Systeme:

Tabelle 40: Bereitstellungsmodelle dezentraler Systeme

System	Bereitstellungsmodell	Abschnitt
[Basisdienst] Authorization Server	Verbindlich	5.4.2.1
[Basisdienst] Dezentrale Policy Infrastruktur	Verbindlich	5.4.2.2
[Basisdienst] API-Gateway	Optional	5.4.2.3
[Basisdienst] SSF-Transmitter-Adapter	Optional	5.4.2.4

Bei verbindlich bereitgestellten Systemen wird eine zentral beschaffte, betreute und vorkonfigurierte Lösung für den dezentralen Betrieb beim Basisdienst bereitgestellt. Basisdienst-

Betreiber setzen diese Lösung ein und sind für deren Betrieb in ihrer lokalen Umgebung verantwortlich.

Bei optional bereitgestellten Systemen wird ebenfalls eine zentral beschaffte, betreute und vorkonfigurierte Lösung bereitgestellt, die produktiv einsetzbar ist. Basisdienst-Betreiber können diese Lösung einsetzen oder alternativ eigene Lösungen verwenden, sofern diese die verbindlichen Schnittstellenspezifikationen und Sicherheitsvorgaben der Zielarchitektur einhalten.

5.4.2.1 Authorization Server (verbindliche Bereitstellung)

Der Basisdienst Authorization Server wird als verbindliche, zentral bereitgestellte Lösung für alle Basisdienste vorgegeben.

Begründung:

Der Authorization Server ist keine rein interne Komponente des Basisdienstes. Er bildet die technische Schnittstelle zum gesamten API-Ökosystem: API Consumer Clients interagieren bei der Client-Registrierung (DCR mit Software Statement Assertion), bei der Token-Anfrage (Client Credentials, Token Exchange) und bei der Authentifizierung (`private_key_jwt`, DPoP) direkt mit dem Authorization Server des Basisdienstes. Das Verhalten des AS wirkt somit unmittelbar auf alle Teilnehmer des föderalen Ökosystems.

Die Sicherheitsvorgaben der Zielarchitektur basieren auf FAPI 2.0 und definieren verbindliche Anforderungen an Client-Authentifizierung, Sender-Constraining und Token-Binding. Diese Vorgaben können sich im Laufe der Zeit ändern – sei es durch neue Versionen des FAPI-Standards, durch Erkenntnisse aus der Betriebserfahrung oder durch geänderte Bedrohungslagen. Eine einheitliche, zentral bereitgestellte Authorization-Server-Lösung ermöglicht es, solche Änderungen schnell, konsistent und zuverlässig im gesamten Ökosystem um-zusetzen, ohne mit jedem Basisdienst-Betreiber einzeln über die Anpassung individueller AS-Implementierungen verhandeln zu müssen.

Darüber hinaus stellt die zentrale Bereitstellung sicher, dass die SSA-basierte dynamische Client-Registrierung, die das Fundament des föderalen Registrierungsmodells bildet, bei allen Basisdiensten interoperabel funktioniert.

5.4.2.2 Dezentrale Policy Infrastruktur (verbindliche Bereitstellung)

Die dezentrale Policy Infrastruktur (PDP und lokaler Attribute Store) wird als verbindliche, zentral bereitgestellte Lösung für alle Basisdienste vorgegeben.

Begründung:

Die dezentrale Policy Infrastruktur muss zwei Kernfunktionen zuverlässig erfüllen, die eine einheitliche Lösung erfordern:

Erstens muss die **Replikationslogik** zwischen der zentralen Policy Infrastruktur (PIP) und dem dezentralen PDP fehlerfrei und konsistent funktionieren. Über diesen Kanal werden Policies, Attributdaten und sicherheitskritische Änderungen (z. B. Client-Sperrungen) an die dezentralen Instanzen ausgeliefert. Dieser Replikationskanal ist kein offener Standard, sondern ein infrastrukturspezifisches Protokoll zwischen zentraler und dezentraler Komponente. Wenn jeder Basisdienst seine eigene PDP-Lösung betreibt, müsste jede diese Replikation individuell und korrekt implementieren. Das Risiko von Inkonsistenzen bei Policy- und Attributdaten – mit unmittelbaren Auswirkungen auf Berechtigungsentscheidungen – wäre zu hoch.

Zweitens muss das **Regelwerk** (die Policy-Evaluierungslogik) ökosystemweit semantisch identisch ausgewertet werden. Policies werden zentral im FöPD verwaltet und über die zentrale Policy Infrastruktur an alle dezentralen PDPs verteilt. Die Wahl der Regelsprache – ob Cedar, Open Policy Agent, DMN oder eine andere Technologie – muss daher einheitlich sein. Eine zentral bereitgestellte Lösung ermöglicht es zudem, das Regelwerk bei Bedarf auszutauschen oder weiterzuentwickeln, ohne dass jeder Basisdienst-Betreiber seine eigene PDP-Implementierung anpassen muss.

Die AuthZEN-Schnittstelle zwischen PEP-Komponenten (AS, API-Gateway) und PDP ist zwar standardisiert und würde grundsätzlich den Einsatz unterschiedlicher PDP-Implementierungen erlauben. Die Gründe für die verbindliche Bereitstellung liegen jedoch nicht in der externen Schnittstelle, sondern in der internen Datenhaltung und Regelkonsistenz.

5.4.2.3 API-Gateway (optionale Bereitstellung)

Für die PEP-Funktion am API-Gateway wird eine zentral beschaffte, betreute und vorkonfigurierte Lösung bereitgestellt. Basisdienst-Betreiber können alternativ eigene Lösungen einsetzen.

Begründung:

Im Gegensatz zum Authorization Server kommuniziert das API-Gateway ausschließlich über standardisierte Schnittstellen mit anderen Infrastrukturkomponenten: Das AuthZEN-Protokoll (OpenID Foundation) dient der Kommunikation mit dem dezentralen PDP, die Token Introspection (RFC 7662) der Kommunikation mit dem lokalen Authorization Server. Die Weiterleitung des API-Aufrufs an den Basisdienst ist eine basisdienst-interne Angelegenheit. Es gibt keine ökosystemweite Außenwirkung, die eine einheitliche Lösung erzwingen würde.

Zudem verfügen viele Basisdienste bereits über eigene Lösungen, die vergleichbare Funktionen übernehmen – etwa das Sichere Anschlusskit (SAK) im Kontext von NOOTS oder Standard-Gateways in der jeweiligen Rechenzentrumsinfrastruktur. Basisdienste, die in Kubernetes-Umgebungen betrieben werden, setzen häufig eigene Ingress Controller ein, die Gateway-Funktionen nativ bereitstellen. Eine Vorschrift, diese bestehenden Lösungen durch ein einheitliches Gateway zu ersetzen, wäre weder verhältnismäßig noch praxisgerecht.

Die Zielarchitektur sieht zwei Bereitstellungsvarianten für die PEP-Logik vor, die dem Basisdienst-Betreiber optional zur Verfügung gestellt werden:

- **Plugin-Variante:** Die PEP-Logik (Token-Introspection-Orchestrierung, AuthZEN-Abfrage) wird als Plugin für das bereitgestellte API-Gateway implementiert. Diese Variante ist für Basisdienste vorgesehen, die kein eigenes Gateway betreiben oder die bereitgestellte Lösung bevorzugen.
- **Sidecar-/Adapter-Variante:** Die PEP-Logik wird als eigenständiger Dienst bereitgestellt, der neben dem bestehenden Gateway des Basisdienstes betrieben werden kann. Das bestehende Gateway leitet den Opaque Access Token und den DPoP-Header an den Adapter weiter. Die Schnittstelle ist bewusst einfach gehalten: Der Adapter übernimmt Introspection und AuthZEN-Abfrage und gibt ein Ergebnis (Zugriff gewährt/verweigert) sowie das JWS Access Token zurück.

Es steht dem Basisdienst-Betreiber darüber hinaus frei, die AuthZEN-Abfrage und die zugehörige Token-Verarbeitung vollständig eigenständig in seiner bestehenden Gateway- oder

Anwendungsinfrastruktur zu implementieren, ohne auf eine der bereitgestellten Varianten zurückzugreifen.

Unabhängig von der gewählten Variante ist die Einhaltung der Sicherheitsvorgaben (u. a. DPoP-Validierung, Token-Prüfung) sowie die korrekte Durchführung der AuthZEN-Abfrage beim dezentralen PDP gemäß Berechtigungskonzept verbindlich. Lediglich die Wahl des Gateway-Produkts ist freigestellt.

5.4.2.4 SSF-Transmitter-Adapter (optionale Bereitstellung)

Für den SSF-Transmitter-Adapter wird eine zentral beschaffte, betreute und vorkonfigurierte Lösung bereitgestellt. Basisdienst-Betreiber können alternativ die SSF-Schnittstelle eigenständig implementieren.

Begründung:

In der Praxis ist der SSF-Transmitter-Adapter in den meisten Fällen kein eigenständiges Deployment-Thema: Wenn ein Basisdienst die verbindlich bereitgestellten Komponenten (Authorization Server, dezentrale Policy Infrastruktur) sowie das optional bereitgestellte API-Gateway einsetzt, ist die SSF-Transmitter-Funktionalität bereits in diesen Komponenten integriert. Die bereitgestellten Systeme erzeugen ihre sicherheits- und betriebsrelevanten Ereignisse gemäß dem föderalen SSF-Eventprofil und übermitteln sie direkt an die zentrale SSF-Monitoring-Infrastruktur.

Der SSF-Transmitter-Adapter als eigenständige Komponente wird nur dann relevant, wenn ein Basisdienst eigene Implementierungen für die API-Gateway-Funktion nutzt und basisdiensteigene Ereignisse (z. B. fachliche Sicherheitsereignisse aus dem Basisdienst selbst) an die SSF-Monitoring-Infrastruktur übermitteln muss. In diesem Fall muss der Betreiber die SET-Daten ohnehin selbst erzeugen und kann die SSF-Schnittstelle (SET-Übermittlung gemäß RFC 9835 im Push- oder RFC 8936 im Poll-Modus) direkt implementieren.

Die verbindliche Vorgabe betrifft in allen Fällen das föderale SSF-Eventprofil (Struktur und Semantik der Security Event Tokens) sowie die Übermittlungsschnittstelle zur zentralen SSF-Monitoring-Infrastruktur. Die Wahl der technischen Umsetzung – integrierter Adapter oder eigenständige Implementierung – ist dem Basisdienst-Betreiber freigestellt.

Da das Shared Signals Framework ein noch relativ neuer Standard ist und die Implementierungserfahrung in der Praxis begrenzt ist, kommt der zentral bereitgestellten Lösung eine

besondere Bedeutung als Beispielimplementierung zu. Sie dient Basisdienst-Betreibern, die eine eigene Implementierung anstreben, als technische Referenz für die korrekte Umsetzung des SSF-Eventprofils und der SET-Übermittlung.

5.5 Betrachtung möglicher Standardlösungen und Nachnutzungsmöglichkeiten für identifizierte Systeme

Das vorliegende Kapitel untersucht für jedes in Kapitel 4.2 identifizierte System, ob primär eine Standardsoftwarelösung oder primär eine Individuallösung eingesetzt werden sollte. „Primär" bedeutet, dass bei einer Individuallösung selbstverständlich für Teilfunktionen (z. B. Workflow-Engines, Datenbanken, Policy-Engines) ergänzend Standardlösungen genutzt werden, und umgekehrt bei einer Standardsoftwarelösung individuelle Konfigurationen und Erweiterungen erforderlich sein können.

Wo Standardsoftwarelösungen in Frage kommen, und geeignete Open-Source-Lösungen existieren, werden diese gemäß Prinzip P-007 (Open-Source-Priorisierung für kritische Komponenten) bevorzugt betrachtet und als mögliche Kandidaten aufgelistet. Eine verbindliche Festlegung auf ein bestimmtes Produkt erfolgt im Rahmen dieses Konzepts nicht. Bei Individuallösungen wird geprüft, ob bestehende Lösungen aus dem Kontext der IT-Planungsrat-Produkte nachgenutzt oder weiterentwickelt werden können (Prinzip P-012, Wiederverwendung und Bündelung vor Neuentwicklung).

5.5.1 Übersicht

Die nachfolgende Tabelle gibt einen Überblick über die Einordnung aller Systeme:

Tabelle 41: Übersicht der Einordnung der Systeme

System	Lösungsansatz	Wesentliche Begründung
Föderales Plattform Directory (FöPD)	Individuallösung mit Standardkomponenten	Fachlich hochspezifisch; OAuth-Server-Komponenten und Workflowsysteme als Standardkomponenten; Nachnutzung IT-PLR-Produkte prüfen
FöPD Identity Provider	Standardsoftware	Etablierter Funktionsumfang in Open-Source-Lösungen verfügbar
Authorization Server für Nutzerzustimmung	Standardsoftware	Gleiche Anforderungen wie FöPD IdP und Basisdienst-AS
Zentrale Policy Infrastruktur	Individuallösung mit Standardkomponenten	Fachspezifische Administration und Propagierung; Policy Engine als Standardkomponente

SSF-Monitoring-Infrastruktur	Individuallösung mit Standardkomponenten	Fachspezifische Integration und Eventprofil; CEP-Engine als Standardkomponente
Transparency Log Infrastruktur	Standardsoftware	Lösungsauswahl bereits erfolgt (siehe Kapitel 5.3.6)
[Basisdienst] Authorization Server	Standardsoftware	Etablierter Funktionsumfang in Open-Source-Lösungen verfügbar
[Basisdienst] Dezentrale Policy Infrastruktur	Individuallösung mit Standardkomponenten	Fachspezifische Replikationslogik; Policy Engine als Standardkomponente
[Basisdienst] API-Gateway	Standardsoftware	Etablierter Funktionsumfang in Open-Source-Lösungen verfügbar
[Basisdienst] SSF-Transmitter-Adapter	Individuallösung mit Open-Source-Bibliotheken	Standard noch jung; Open-Source-Bibliotheken als Grundlage verfügbar

5.5.2 Systeme mit primärem Standardsoftwareeinsatz

5.5.2.1 FöPD Identity Provider

Der FöPD Identity Provider dient als Vertrauensanker für die Nutzeridentitäten innerhalb der Plattform. Die Anforderungen – OpenID Connect Provider, Nutzerverwaltung, Authentifizierung, Föderierung mit externen Identity Providern (BundID, MUK) – werden von etablierten Open-Source-Identity-Plattformen abgedeckt.

Spezifisch für den Einsatz im FöPD ist die Anforderung einer nativen SCIM-Schnittstelle (RFC 7644), um Identitäts- und Organisationsdaten standardisiert für andere Systemkomponenten (insbesondere die zentrale Policy Infrastruktur) bereitstellen zu können.

Mögliche Open-Source-Kandidaten:

Tabelle 42: Open Source Kandidaten FöPD Identity Provider

Lösung	Lizenz	Governance	Hinweise
Janssen (Gluu/Linux Foundation)	Apache 2.0	Linux Foundation	Nativer SCIM-Support, FAPI-zertifiziert, FIDO-Unterstützung, Open-Banking-Erfahrung. Enterprise Support über Gluu Flex.
Keycloak (Red Hat/CNCF)	Apache 2.0	CNCF	Breite Community, umfangreiche IdP-Funktionalität, Red Hat Enterprise Support. SCIM-Support ab Version 26.6

			geplant (Stand: März 2026 noch experimentell).

5.5.2.2 Authorization Server für Nutzerzustimmung

Der zentrale Authorization Server für Nutzerzustimmung stellt API-unspezifische Opaque Access Tokens mit Nutzerautorisierung aus (Authorization Code Flow mit PKCE). Die Anforderungen an dieses System entsprechen im Kern denen des FöPD Identity Providers und des Basisdienst Authorization Servers: FAPI-2.0-Konformität, DPoP-Unterstützung, SSA-basierte dynamische Client-Registrierung (RFC 7591) und `private_key_jwt`-Authentifizierung.

Mögliche Open-Source-Kandidaten: Siehe Kapitel 5.5.2.1 (FöPD Identity Provider) und Kapitel 5.5.2.3 (Basisdienst Authorization Server). Aus Gründen der betrieblichen Konsistenz und des Know-how-Aufbaus ist der Einsatz derselben Lösung für alle drei Rollen (FöPD IdP, zentraler AS, dezentraler AS) naheliegend.

5.5.2.3 [Basisdienst] Authorization Server

Der Basisdienst Authorization Server wird als verbindliche, zentral bereitgestellte Lösung für alle Basisdienste vorgegeben (siehe Kapitel 5.4.2.1). Die Anforderungen umfassen FAPI-2.0-Konformität, SSA-basierte dynamische Client-Registrierung (RFC 7591), `private_key_jwt`-Authentifizierung (ADR-002), DPoP-Sender-Constraining (ADR-003), Token Exchange (RFC 8693) und Token Introspection (RFC 7662).

Die SSA-basierte dynamische Client-Registrierung ist eine Kernanforderung der Zielarchitektur und bildet das Fundament des föderalen Registrierungsmodells (ADR-004). Sie ist damit ein wesentliches Differenzierungsmerkmal bei der Bewertung von Kandidaten.

Mögliche Open-Source-Kandidaten:

Tabelle 43: Open Source Kandidaten Authorization Server

Lösung	Lizenz	Governance	Hinweise
Janssen (Gluu/Linux Foundation)	Apache 2.0	Linux Foundation	Nativer SSA-Support in der DCR, FAPI-zertifiziert (FAPI 1 Advanced Final), DPoP und `private_key_jwt` unterstützt, SCIM nativ, Open-Banking-Produktionserfahrung (u. a. Brasilien). Enterprise Support über Gluu Flex.

			Erweiterbarkeit über Interception Scripts (Java/Python).
Keycloak (Red Hat/CNCF)	Apache 2.0	CNCF	FAPI 2.0 Final und DPoP ab Version 26.4 vollständig unterstützt. Token Exchange (RFC 8693) ab Version 26.2 offiziell unterstützt. Sehr große Community und breites Dienstleister-Ökosystem. SSA-Support in der DCR fehlt (offenes Enhancement-Issue #40555, Stand: März 2026).

Gemäß Prinzip P-007 sollte bei der Auswahl darauf geachtet werden, dass eine dokumentierte Fallback-Option definiert wird. Sollte eine der Lösungen die fehlenden Funktionen nachreichen (z. B. Keycloak mit SSA-Support), ist eine Neubewertung angezeigt.

5.5.2.4 [Basisdienst] API-Gateway

Das API-Gateway wird als optionale, zentral bereitgestellte Lösung angeboten (siehe Kapitel 5.4.2.3). Die Anforderungen umfassen neben den projektspezifischen PEP-Funktionen (DPoP-Validierung, Token Introspection, AuthZEN-Abfrage) auch klassische Gateway-Funktionen wie Routing, TLS-Terminierung, Rate Limiting und Observability.

Da die projektspezifischen PEP-Funktionen (insbesondere die AuthZEN-Abfrage) bei keinem marktgängigen Gateway nativ verfügbar sind, ist die Plugin-Architektur des Gateways ein wesentliches Bewertungskriterium: Sie muss es ermöglichen, die fehlenden Funktionen effizient als Custom Plugins zu ergänzen. Auch die DPoP-Validierung (RFC 9449) wird von den meisten Gateways nicht nativ unterstützt, ist aber über Custom Plugins umsetzbar, sofern die Plugin-Architektur Zugriff auf Request-Header und JWT-Bibliotheken bietet. Darüber hinaus ist das Dienstleister-Ökosystem im DACH-Raum relevant, da die Basisdienst-Betreiber für den Betrieb des Gateways auf externe Unterstützung angewiesen sein können.

Ein weiterer Aspekt ist die Einsatzflexibilität: Das Gateway muss sowohl in klassischen Rechenzentrumsumgebungen als auch in Kubernetes-Umgebungen betrieben werden können. In Kubernetes-Umgebungen kann der Ingress Controller des Basisdienstes die klassischen Gateway-Funktionen übernehmen; in diesem Fall sind die projektspezifischen PEP-Funktionen entweder als Plugin im Ingress Controller oder als Sidecar-Adapter bereitzustellen.

Mögliche Open-Source-Kandidaten:

Tabelle 44: Open Source Kandidaten API-Gateway

Lösung	Lizenz	Governance	Hinweise
Apache APISIX	Apache 2.0	Apache Software Foundation	Leichtgewichtig (NGINX/OpenResty + etcd), Hot-Reload, Plugin-Architektur in Lua (nativ) sowie Java/Go/Python (via Plugin Runner). ASF-Governance (Vendor-neutral). Enterprise Support über API7.ai. DPoP über Custom Plugin umsetzbar.
Tyk	MPL 2.0	Tyk Technologies Ltd (UK)	Go-basiert, leichtgewichtig (Redis als einzige Dependency), Plugin-Architektur in Go (nativ), Python, JavaScript und gRPC. Open-Banking-Erfahrung, DPoP-Unterstützung im Rahmen des FAPI Accelerators. Einschränkung: MPL-2.0-Lizenz ist restriktiver als Apache 2.0; Single-Vendor-Projekt. Dashboard nur in Enterprise-Edition.
KrakenD	Apache 2.0 (Community)	KrakenD Inc. (ES), Lura-Framework bei der Linux Foundation	Go-basiert, stateless, extrem leichtgewichtig (kein DB-Dependency), hohe Performance. Plugin-Architektur in Go, Lua und Google CEL. Enterprise Support über KrakenD Inc. DPoP über Custom Plugin umsetzbar.
Gravitee.io	Apache 2.0 (Kern)	Gravitee Source SAS (FR)	Java-basiert, umfangreiche API-Management-Plattform mit Developer Portal. Plugin-Architektur in Java (Maven). Europäischer Anbieter. DPoP über Custom Policy umsetzbar. Einschränkung: Schwergewichtig (Java + DB + Elasticsearch), für den reinen PEP-Sidecar-Einsatz überdimensioniert.

5.5.2.5 Transparency Log Infrastruktur

Die Lösungsauswahl für die Transparency Log Infrastruktur wurde bereits in Kapitel 5.3.6 getroffen. Als Lösung wurde Tessera (transparency-dev/tessera) identifiziert. Für die Begründung und Details der Entscheidung wird auf das genannte Kapitel verwiesen.

5.5.3 Systeme mit primärem Individuallösungsansatz

5.5.3.1 Föderales Plattform Directory (FöPD)

Das FöPD ist die zentrale Verwaltungs- und Registrierungsplattform der föderalen API-Autorisierungsinfrastruktur. Es umfasst hochspezifische Fachfunktionen wie die Verwaltung von Plattformangeboten, die Registrierung von Organisationen und Software, die Ausstellung von Software Statement Assertions, ein Self-Service-Portal für Basisdienst-Nutzer und Betreiber sowie Workflow-Funktionalitäten für Antrags- und Freigabeprozesse. Dieser Funktionsumfang ist fachlich so spezifisch, dass keine Standardsoftwarelösung ihn vollständig abdecken kann. Wesentliche Funktionsblöcke können jedoch durch Standardkomponenten abgedeckt werden:

- **OAuth-Server-Komponenten:** Die Verwaltung von Software Clients und die Ausstellung von Software Statement Assertions (SSA) sind Kernfunktionen des OAuth-Ökosystems. Für diese Funktionen können Komponenten einer OAuth-Server-Lösung (z. B. Janssen) nachgenutzt werden, die SSA-Generierung, Client-Metadaten-Verwaltung und DCR-Unterstützung nativ bereitstellen.
- **Workflowsysteme:** Die Antrags- und Freigabeprozesse (z. B. Registrierung von Organisationen, Freigabe von Plattformangeboten) können durch ein Workflow-Management-System als Standardkomponente unterstützt werden.

Gemäß Prinzip P-012 (Wiederverwendung und Bündelung vor Neuentwicklung) ist zu prüfen, ob bestehende Lösungen aus dem Kontext der IT-Planungsrat-Produkte als Grundlage nachgenutzt werden können. Als präferierter Nachnutzungskandidat wird das **Self-Service-Portal von FIT-Connect** betrachtet, das bereits vergleichbare Registrierungs- und Verwaltungsfunktionen für den föderalen Kontext bereitstellt und somit wesentliche Teile der benötigten Portalfunktionalität abdecken könnte.

5.5.3.2 Zentrale Policy Infrastruktur

Die zentrale Policy Infrastruktur ist die kanonische Quelle für alle aktiven Policies und stellt die Datengrundlage für die dezentralen Policy-Infrastrukturen bereit. Sie umfasst fachspezifische Funktionen wie Policy-Administration, Verwaltung des Attributkatalogs, Propagierung von Policies und Attributdaten an die dezentralen PDPs (PIP-Funktion) sowie die Integration mit dem FöPD für Kataloginformationen.

Die Kombination aus fachspezifischer Administrationsoberfläche und dem proprietären Replikationsprotokoll zum dezentralen PDP macht eine Individuallösung erforderlich. Für die Policy-

Evaluierungslogik im Kern (Regelauswertung, Combining Algorithm) können jedoch Standardkomponenten als Policy Engine eingesetzt werden:

Mögliche Standardkomponenten für die Policy Engine:

Tabelle 45: Kandidaten für Policy Engine

Lösung	Typ	Hinweise
Cedar (AWS/Apache 2.0)	Policy-Sprache und Evaluierungs-Engine	Ausdrucksstarke Regelsprache, formale Verifikation, optimiert für Autorisierungsentscheidungen, nicht generisch.
Open Policy Agent (OPA/CNCF)	Policy Engine	CNCF-Projekt, breite Adoption, Rego als Regelsprache, sehr generisch und ausdrucksstark, eingebaute Management-API, gut geeignet für attributbasierte Zugriffskontrolle.
DMN-Engines (z. B. Camunda DMN, Drools)	Entscheidungs-Engine	Herstellerunabhängiger Standard (OMG Decision Model and Notation). Im Autorisierungskontext unüblicher als OPA/Cedar, aber grundsätzlich geeignet für regelbasierte Entscheidungen. Vorteil: standardisierte, toolgestützte Modellierung von Entscheidungstabellen.

Die Entscheidung für eine konkrete Policy Engine ist im Rahmen dieses Konzepts noch nicht getroffen und wird in einer nachgelagerten Detailkonzeption adressiert.

5.5.3.3 [Basisdienst] Dezentrale Policy Infrastruktur

Die dezentrale Policy Infrastruktur wird als verbindliche, zentral bereitgestellte Lösung vorgegeben (siehe Kapitel 5.4.2.2). Sie umfasst den Policy Decision Point (PDP) und einen lokalen Attribute Store. Im Unterschied zur zentralen Policy Infrastruktur liegt der Schwerpunkt nicht auf der Policy-Administration, sondern auf der Replikation und lokalen Vorhaltung von Policies und Attributdaten sowie deren effizienter Auswertung zur Laufzeit.

Die Replikationslogik (Empfang von Policies und Attributdaten über den PIP-Kanal) ist infrastrukturspezifisch und erfordert eine Individuallösung. Für die Policy-Evaluierungslogik wird dieselbe Standardkomponente (Policy Engine) eingesetzt wie in der zentralen Policy Infrastruktur, um die semantische Konsistenz der Regelauswertung sicherzustellen. Siehe Kapitel 5.5.3.2 für die möglichen Standardkomponenten.

5.5.3.4 SSF-Monitoring-Infrastruktur

Die SSF-Monitoring-Infrastruktur aggregiert, korreliert und bewertet sicherheitsrelevante Ereignissignale aller Basisdienste und zentralen Komponenten. Sie implementiert das Shared Signals Framework (SSF) der OpenID Foundation als zentralen Event-Receiver und -Broker.

Anforderungsanalyse

Die SSF-Monitoring-Infrastruktur muss als zentraler SSF-Receiver Events von einer Vielzahl dezentraler Transmitter (Basisdienst-Komponenten und zentrale Infrastrukturkomponenten) empfangen, diese über Subjekte und Basisdienste hinweg korrelieren und aggregierte Risikosignale ableiten. Dabei kommen neben den standardisierten CAEP- und RISC-Event-Typen (z. B. Session Revoked, Credential Change, Device Compliance Change) auch infrastrukturspezifische Event-Typen zum Einsatz, die im föderalen SSF-Eventprofil definiert werden – etwa Token-Ausstellungen, abgelehnte Token-Requests, Policy-DENY-Entscheidungen, Replikationsfehler oder Anomaliemuster im Request-Volumen.

Die SSF-Spezifikation ist grundsätzlich erweiterbar: Event-Typen werden über URIs identifiziert, und das Framework erlaubt die Definition neuer Event-Typen jenseits des CAEP/RISC-Katalogs. Die Frage ist daher, ob verfügbare Produkte diese Erweiterbarkeit unterstützen.

Anbieter mit SSF-Receiver-Funktionalität:

Tabelle 46: Anbieter mit SSF-Funktionalität

Anbieter	Hinweise
SGNL	SSF-Receiver für CAEP- und RISC-Events, empfängt Events per Push-Delivery von externen Quellen und integriert sie in Policy-Entscheidungen.
SailPoint (Identity Security Cloud)	SSF-Receiver und -Transmitter, unterstützt u. a. Device Compliance, Risk Level Change, Token Claims Change und Session Revoked Events. Workflow-basierte Reaktion auf empfangene Events.
Okta	Identity Threat Protection mit SSF-Receiver- und Transmitter-Funktion für CAEP- und RISC-Ereignisse.
Google Workspace	SSF-Receiver für CAEP-Signale (Closed Beta, Stand: März 2026).

Bewertung für den Einsatz in der Zielarchitektur

Die verfügbaren kommerziellen Lösungen sind auf die standardisierten CAEP/RISC-Event-Typen ausgerichtet. Die föderale API-Autorisierungsinfrastruktur benötigt jedoch die Verarbeitung infrastrukturspezifischer Custom-Event-Typen, die im föderalen SSF-Eventprofil definiert werden. Es ist zum aktuellen Zeitpunkt nicht sichergestellt, dass die genannten Produkte Custom-Event-Typen jenseits des CAEP/RISC-Katalogs empfangen, interpretieren und in ihre Korrelationslogik einbeziehen können. Darüber hinaus erfordert die Zielarchitektur eine Korrelation über infrastrukturspezifische Subjekte (API-Consumer-Clients, Software Statements, Organisationen), die von den auf Identity-Szenarien zugeschnittenen kommerziellen Produkten möglicherweise nicht abgebildet werden.

Für die SSF-Monitoring-Infrastruktur ist daher eine Individuallösung mit bedeutenden Standardkomponenten-Anteilen vorgesehen. Die Architektur gliedert sich in drei Schichten, für die jeweils Open-Source-Standardkomponenten oder -Bibliotheken verfügbar sind:

Schicht 1 – SSF-Protokoll-Layer (Event-Empfang, Stream-Management, SET-Validierung)

Für den SSF-Protokoll-Layer existieren Open-Source-Bibliotheken, die Stream-Aufbau, SET-Empfang und Signaturprüfung bereitstellen:

Tabelle 47: Open Source Bibliotheken zum SSF-Protokoll

Projekt	Sprache	Hinweise
SGNL caep.dev-receiver	Go	SSF-Receiver-Bibliothek mit Stream-Management und Poll-Delivery. Bereitgestellt von SGNL in Kooperation mit der OpenID Foundation.
duo-labs/sharedsignals	Python	Transmitter- und Receiver-Beispiele von Cisco (sharedsignals.guide). Proof-of-Concept, nicht produktionsreif, aber als Lernressource geeignet.
nevzatcirak/sharedsignals	Java (Spring Boot)	SSF-Referenzimplementierung mit Push- und Poll-Delivery, Stream-Management und Event-Buffering.
identitytailor/keycloak-ssf-support	Java	Keycloak-Extension als SSF-Receiver mit pluggable Event Handlers.

Schicht 2 – Event-Akkumulation, Korrelation und Risikobewertung

Für die zeitliche Aggregation, Subjekt-übergreifende Korrelation und Risikobewertung kann eine Open-Source-CEP-Engine (Complex Event Processing) als Standardkomponente eingesetzt werden. CEP-Engines sind darauf ausgelegt, Muster über Event-Streams in Echtzeit zu erkennen, zeitfensterbasierte Aggregationen zu berechnen und schwellwertbasierte Risikobewertungen abzuleiten. Damit decken sie sowohl die Korrelation als auch die Risikowert-Berechnung in einer Komponente ab.

Tabelle 48: Open Source Kandidaten CEP-Engine

Lösung	Sprache	Governance	Hinweise
Apache Flink (CEP-Library)	Java	Apache Software Foundation	Ausgereiftes Stream-Processing-Framework mit eingebauter CEP-Bibliothek für Muster-Erkennung über Event-Streams. Skalierbar, produktionserprobt bei Unternehmen wie PayPal und Booking.com für Security-Event-Korrelation.
Siddhi	Java/Python	Apache 2.0 (WSO2)	Speziell für CEP konzipierte Engine mit SQL-ähnlicher Abfragesprache. Bietet Aggregationen über Zeit- und Session-Windows, Pattern-Analyse, Anomalieerkennung und schwellwertbasiertes Alerting. Kann als eingebettete Bibliothek oder als Microservice betrieben werden.
Esper	Java	GPL 2.0	Eines der ausgereiftesten CEP-Engines, seit 2006 im Einsatz. Eigene Event Processing Language (EPL). Einschränkung: GPL-Lizenz ist für den Einsatz in der öffentlichen Verwaltung zu prüfen.

Schicht 3 – Signal-Weiterleitung

Die SSF-Monitoring-Infrastruktur fungiert ihrerseits als SSF-Transmitter gegenüber den dezentralen Policy-Infrastrukturen und angeschlossenen SIEM-Systemen. Für die Transmitter-Logik können dieselben SSF-Bibliotheken genutzt werden wie in Schicht 1.

Individueller Anteil

Der individuelle Entwicklungsanteil liegt in der Integration dieser drei Schichten, der Definition und Implementierung des föderalen SSF-Eventprofils (Custom-Event-Typen jenseits des CAEP/RISC-Katalogs) sowie der Konfiguration der Korrelationsregeln und Risikoschwellwerte für das spezifische Bedrohungsmodell der föderalen API-Autorisierungsinfrastruktur.

Open-Source-Bereitstellung der Eigenentwicklung

Die im Rahmen des Projekts entwickelte SSF-Monitoring-Lösung wird als Open Source nachnutzbar bereitgestellt und der Community zur Weiterentwicklung übergeben. Diese Investition wird durch personelle und finanzielle Mittel begleitet, um eine nachhaltige Weiterentwicklung sicherzustellen. Da produktionsreife Open-Source-Lösungen im SSF-Bereich bislang fehlen, kann diese Bereitstellung einen wesentlichen Beitrag zum SSF-Ökosystem leisten und auch anderen Anwendern im öffentlichen Sektor als Grundlage dienen.

5.5.3.5 [Basisdienst] SSF-Transmitter-Adapter

Der SSF-Transmitter-Adapter übersetzt basisdienst-interne Ereignisse in das föderale SSF-Eventprofil und übermittelt sie an die zentrale SSF-Monitoring-Infrastruktur (siehe Kapitel 5.4.2.4). In den verbindlich bereitgestellten Komponenten (Authorization Server, dezentrale Policy Infrastruktur) sowie im optional bereitgestellten API-Gateway ist die SSF-Transmitter-Funktionalität integriert.

Für Basisdienste, die eigene Implementierungen der API-Gateway-Funktionen einsetzen und basisdiensteigene Ereignisse übermitteln müssen, wird eine eigenständige Implementierung des SSF-Transmitter-Adapters erforderlich. Da das Shared Signals Framework ein noch junger Standard ist (Final-Spezifikationen im September 2025 verabschiedet), ist die Verfügbarkeit produktionsreifer Implementierungen eingeschränkt.

Es existieren jedoch Open-Source-Bibliotheken, die als Grundlage für eine Implementierung dienen können:

Verfügbare Open-Source-Bibliotheken und Referenzimplementierungen:

Tabelle 49: Open Source Kandidaten SSF-Transmitter-Adapter

Projekt	Sprache	Typ	Hinweise
SGNL caep.dev-receiver	Go	SSF-Receiver-Bibliothek	Bereitgestellt von SGNL in Kooperation mit der OpenID Foundation. Kostenlose Online-Test-Infrastruktur unter caep.dev.
duo-labs/sharedsignals	Python	Transmitter- und Receiver-Beispiele	Proof-of-Concept von Cisco (sharedsignals.guide). Dockerisiert, nicht produktionsreif, aber als Lernressource geeignet.
ne-vzatcirak/sharedsignals	Java (Spring Boot)	SSF-Referenzimplementierung	Transmitter-Implementierung mit Push- und Poll-Delivery, Stream-Management und Event-Buffering.
identitytailor/keycloak-ssf-support	Java	Keycloak-Extension	SSF-Receiver für Keycloak mit pluggable Event Handlers. Transmitter-Support in Entwicklung.
mjovanc/sigshare	Rust	SSF-Bibliothek	In aktiver Entwicklung.

Für die zentral bereitgestellte Implementierung des SSF-Transmitter-Adapters können diese Bibliotheken als Ausgangspunkt genutzt werden. Die bereitgestellte Lösung dient gemäß Kapitel 5.4.2.4 zugleich als Beispielimplementierung für Basisdienst-Betreiber, die eine eigenständige Integration anstreben.

Analog zur SSF-Monitoring-Infrastruktur (Kapitel 5.5.3.4) wird auch die SSF-Transmitter-Adapter-Implementierung als Open Source bereitgestellt und der Community zur Weiterentwicklung übergeben.

5.6 Middleware als bevollmächtigter API-Consumer

Middleware-Systeme nehmen in der behördlichen IT-Praxis eine zentrale Rolle ein: Sie verbinden Fachverfahren, Dokumentenmanagementsysteme und weitere Anwendungen mit zentralen Diensten, gleichen Schnittstellenunterschiede aus und bilden Arbeitsabläufe über Systemgrenzen hinweg ab. Die föderale API-Autorisierungsinfrastruktur berücksichtigt diese Realität – Middleware ist kein Sonderfall, der gesonderte Architekturmechanismen erfordert, sondern ein regulärer Anwendungsfall des allgemeinen Client-Modells.

Grundprinzip

Eine Middleware tritt gegenüber der Infrastruktur wie jeder andere API-Consumer auf: als registrierter Software-Client mit eigenem Software Statement und eigenem Schlüsselmaterial. Die Besonderheit liegt nicht in einer privilegierten Stellung der Middleware, sondern in ihrer Betriebsweise: Sie verwaltet intern mehrere Identitäten – idealtypisch: eine pro angebundener Fachanwendung – und stellt Anfragen stets im Kontext der jeweiligen Fachanwendung. Diese Form von "Identity-Multiplexing" entspricht dem etablierten Muster der Mandantenfähigkeit ausgereifter Middleware-Lösungen und erfordert keine grundlegende Architekturüberarbeitung.

Ein dynamischer Identitätswechsel zur Laufzeit – bei dem eine Middleware mit einem einzigen Software Statement flexibel verschiedene Fachverfahrenskontexte annimmt – ist in diesem Konzept bewusst nicht vorgesehen. Diesem würde die klare Zurechenbarkeit von Zugriffen unterlaufen und dem Zero-Trust-Prinzip widersprechen. Die Identitätsauflösung erfolgt bevorzugt im Registrierungsprozess, nicht zur Laufzeit.

Registrierung und Onboarding

Das Onboarding von Middleware-Systemen erfolgt bevorzugt über den allgemeinen Software-Statement-Registrierungsprozess des FöPD. Der Standardfall ist die Registrierung im Auftrag der Fachverantwortlichen Stelle bzw. der jeweiligen Behörde – mit einem eigenständigen Software Statement pro angebundener Fachanwendung.

Hierbei sind folgende Varianten der Ausführung möglich:

Registrierung im Auftrag einer Behörde (Standardfall): Eine Middleware registriert am FöPD ein Software Statement für eine angebundene Fachanwendung. Die zuständige Stelle/Behörde erhält hierzu eine Benachrichtigung und genehmigt den Antrag über einen definierten Workflow.

Nach Freigabe wird das Statement signiert; die Middleware kann es anschließend nutzen, um die Fachanwendung gegenüber den Basisdiensten zu vertreten. Mit der Genehmigung übernimmt die zuständige Stelle/Behörde die organisatorische Verantwortung für das ausgestellte Statement. Die zuständige Stelle, in deren Namen das Software Statement genehmigt wurde, wird verantwortliche Eigentümerin der Vertrauensbeziehung. Die Beziehung ist für alle Parteien einsehbar und kann unilateral invalidiert werden.

Weiterhin können dem Software-Statement fachverfahrens-relevante Attribute zugewiesen werden, auf deren Grundlage die Software entsprechende Basisdienste nutzen kann.

Dieses Muster stellt sicher, dass jede Fachanwendung mit einer eigenständigen, klar abgegrenzten Identität und einem minimalen Berechtigungsumfang operiert. Monitoring, Risikobewertung und Zurechenbarkeit von Zugriffen bleiben auf Ebene der einzelnen Fachanwendung erhalten.

Registrierung im eigenen Namen (Ausnahmefall): Eine Middleware kann ausnahmsweise als eigenständiger Software-Client im FöPD registriert werden, wenn sie nachweislich ausschließlich im Auftrag einer einzigen Fachanwendung und Behörde handelt und keine weiteren Anbindungen zu erwarten sind. Dieser Weg ist eng zu begrenzen: Eine Middleware, die im eigenen Namen registriert ist und sukzessive Attribute weiterer Fachanwendungen resp. Behörden akkumuliert, verletzt das Least-Privilege-Prinzip und erschwert Monitoring und Risikobewertung erheblich. Es ist daher nicht als Grundlage für den Regelbetrieb geeignet.

> **Hinweis:** Die Registrierung von Software Statements am FöPD durch Middleware-Betreiber soll über eine dedizierte API in der FöPD-Administrationsschicht automatisiert erfolgen. Der zugehörige Freigabeprozess – die Genehmigung durch die vertretene Behörde – ist dabei so zu gestalten, dass die Delegationsbeziehung zwischen Middleware und Fachanwendung nachvollziehbar dokumentiert wird. Die konkrete Ausgestaltung ist Teil des Implementierungskonzepts.

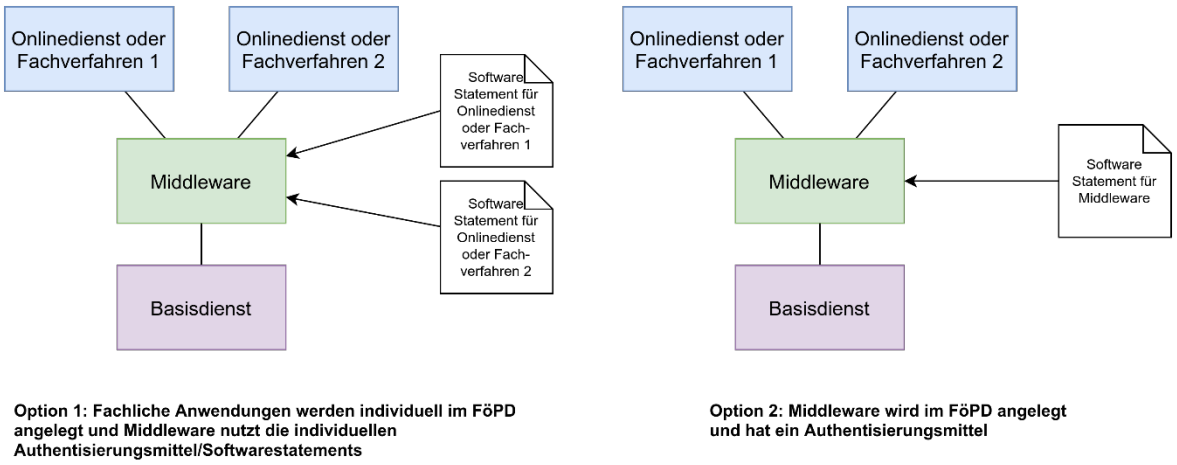
Delegation und Nachvollziehbarkeit

Die Beziehung zwischen Middleware und vertretener Fachanwendung ist keine Dimension des Berechtigungsmodells, sondern eine Provenienz-Information des Software Statements: Sie dokumentiert, wer das Statement eingereicht hat und wer dessen Registrierung im Namen der Behörde genehmigt hat. Diese Information wird im Rahmen des Registrierungsprozesses im FöPD erfasst und ist Bestandteil des Audit-Trails.

Der PDP wertet ausschließlich aus, was das Statement über den Client aussagt: über Attribute, Berechtigungen, organisatorische Zugehörigkeit. Die Herkunft und Delegations-Beziehung des Statements ist für die Berechtigungsentscheidung ohne Belang. Mit der Genehmigung durch die fachverantwortliche Stelle/Behörde übernimmt diese die organisatorische Verantwortung für das ausgestellte Statement; die Middleware handelt auf dieser Grundlage als bevollmächtigter Einreicher, nicht als eigenständiger Berechtigungsakteur.

Das Berechtigungsmodell wird durch die Einbindung von Middleware damit nicht komplexer. Die gesamte Vertrauensherstellung ist im Registrierungsprozess gekapselt und zur Laufzeit bereits aufgelöst.

Die nachfolgende Abbildung illustriert die zwei grundlegenden Varianten zur Identitätsbindung von Middleware durch entsprechende Software Statements:



Option 1: Fachliche Anwendungen werden individuell im FöPD angelegt und Middleware nutzt die individuellen Authentisierungsmittel/Softwarestatements

Option 2: Middleware wird im FöPD angelegt und hat ein Authentisierungsmittel

Abbildung 66: Varianten zur Identitätsbindung von Middleware

Im Falle der „Option 1" agiert die Fachanwendung (Client) in mehreren verschiedenen Nutzungskontexten; beispielsweise um einmal für „Behörde 1" am „Fachverfahren 1" und für „Behörde 2" am „Fachverfahren 2" teilzunehmen, je repräsentiert durch ein Software Statement. Gemäß dem Prinzip der Mandantentrennung wird sichergestellt, dass die Middleware das für den jeweiligen Kontext korrekte Software Statement verwendet, um am Fachverfahren teilzunehmen und sich als bevollmächtigte Akteur zu authentifizieren. Dies kann beispielsweise durch ein entsprechendes Deployment der Middleware erreicht werden (eine Laufzeit-Instanz + Software Statement je Kontext, oder durch eine interne Programmlogik, die zur Laufzeit das jeweils korrekte Softwarestatement verwendet (Mandantentrennung zur Laufzeit).

Bei „Option 2" wird die Middleware nur durch ein einzelnes Software Statement repräsentiert und nutzt diese Identität zur Laufzeit, um an mehreren verschiedenen Fachverfahren teilzunehmen. Das Software Statement repräsentiert die Middleware selbst, eine Unterscheidung verschiedener Identitätskontexte (Fachanwendung, Behörde) ist so nicht eindeutig möglich.

Grundsätzlich ist Option 1 zu bevorzugen, da hierbei der Identitätskontext des handelnden Akteurs für alle beteiligten Komponenten klar ersichtlich und unterscheidbar ist. Option 2 ist zu vermeiden und nur in begründeten Ausnahmefällen zulässig.

Integrationsaufwand

Der Integrationsaufwand für bestehende Middleware-Systeme ist überschaubar, sofern diese bereits mandantenfähig ausgestaltet ist. Das wesentliche Credential-Format ändert sich von einem Client-Secret auf ein Schlüsselpaar für Private-Key-JWT – die strukturelle Anforderung, pro Mandanten isolierte Credentials zu verwalten, sollte in modernen Middleware-Lösungen bereits bestehen.

Neu zu implementieren ist hingegen die Dynamic Client Registration gemäß RFC 7591: Die Middleware muss für jede vertretene Fachanwendung/Behörde am Authorization Server des jeweiligen Basisdienstes eine Client-ID beantragen (i. d. R. einmalig pro Software Statement je Basisdienst). Dies erfolgt unter Vorlage des vom FöPD ausgestellten Software Statements und ist ein Schritt, den bestehende Middleware-Lösungen heute typischerweise nicht abbilden. Die Unterstützung dieses Registrierungsflows ist daher bei der Integrationsplanung miteinzuplanen.

Zusammengefasst ergibt sich für Middleware-Betreiber folgendes Bild:

- **Mandantenfähigkeit und Credential-Verwaltung:** in modernen Lösungen bereits vorhanden; Anpassung auf Schlüsselpaar-basierte Authentisierung erforderlich bzw. nachzuprüfen
- **Registrierung im FöPD:** erfolgt über die FöPD-API mit anschließendem Freigabe-Workflow; kein gesondertes Onboarding-Verfahren erforderlich
- Dynamic Client Registration am Basisdienst-AS (RFC 7591): neu zu implementierende Funktion
- **Token Request am Basisdienst-AS (RFC 7523 & 9449)**: Anpassung der Implementierung für bestehenden Prozess

6 Transitionsbetrachtung

Die Einführung der Föderalen API-Autorisierungsinfrastruktur betrifft eine Vielzahl bestehender und in Entwicklung befindlicher föderaler Basisdienste. Dieses Kapitel gibt einen Überblick über die identifizierten betroffenen Basisdienste (Kapitel 6.1) und benennt mögliche Strategien für die schrittweise Anbindung an die Zielarchitektur (Kapitel 6.2). Eine konkrete Roadmap mit verbindlichen Zeitpunkten und Reihenfolgen kann erst aus der bilateralen Abstimmung mit den Plattformverantwortlichen Stellen abgeleitet werden und ist nicht Teil dieses Dokuments.

6.1 Übersicht föderaler Basisdienste

6.1.1 Definition und Vorgehen

Als föderaler Basisdienst im Sinne dieser Übersicht gelten Dienste, die drei Hartkriterien erfüllen: Erstens Querschnittlichkeit, das heißt der Dienst wird nicht für eine einzelne Fachdomäne, sondern domänenübergreifend für viele Verwaltungsleistungen genutzt. Zweitens föderale Mitnutzung durch mehrere Ebenen oder Länder, also nicht nur durch eine einzelne Behörde oder ein einzelnes Land. Drittens API-Charakter, das heißt der Dienst stellt zumindest eine maschinenlesbare Schnittstelle für die Anbindung anderer Systeme bereit. Entscheidend ist, dass die API tatsächlich föderal angeboten und genutzt wird, nicht nur konzeptionell vorgesehen ist.

Die Übersicht in Kapitel 6.1.3 fasst alle in Bund und Ländern bekannten oder in der Entwicklung befindlichen Basisdienste zusammen. Eine konsolidierte autoritative Liste föderaler Basisdienste existiert nach Kenntnis der Autoren nicht; die Übersicht wurde auf Basis öffentlich zugänglicher Quellen wie Architekturkonzepten, Beschlüssen des IT-Planungsrats und Produktbeschreibungen der bereitstellenden Stellen zusammengetragen. Sie ist daher nur als Arbeitsgrundlage zu verstehen, die im weiteren Vorhabensverlauf gemeinsam mit den Plattformverantwortlichen Stellen zu validieren und zu ergänzen ist.

6.1.2 Angrenzende Domänen mit Anschlusspotenzial

Über den hier betrachteten Kreis hinaus existieren angrenzende Domänen mit eigener Governance und etablierten Infrastrukturen, die zwar nicht im engeren Sinne als föderale Basisdienste der Verwaltungsdigitalisierung gelten, aber Anschlusspotenzial an die hier beschriebene Zielarchitektur aufweisen.

Im Bereich der **Sozialversicherung** koordiniert die Informationstechnische Servicestelle der Gesetzlichen Krankenversicherung (ITSG) zentrale Verfahren des elektronischen

Datenaustauschs auf Grundlage der Datenerfassungs- und -übermittlungsverordnung (DEÜV). Hinzu kommt das SV-Meldeportal als Online-Dienst für Arbeitgeber. Diese Verfahren folgen einer eigenen Governance der Sozialversicherungsträger und sind nicht Teil der OZG-Basisdienste-Landschaft, weisen aber inhaltliche Berührungspunkte zur Wirtschaftsverwaltung auf.

Im Bereich der **Justiz** liegt die Verantwortung für die digitale Infrastruktur bei der Bund-Länder-Kommission für Informationstechnik in der Justiz (BLK). Etablierte Komponenten umfassen das Elektronische Gerichts- und Verwaltungspostfach (EGVP) als Zustellinfrastruktur, den Verzeichnisdienst SAFE für die Auflösung der Zustellkontakte, den XÖV-Standard XJustiz für die strukturierte Datenübermittlung sowie das Transportprotokoll OSCI. Konkrete Fachanwendungen mit API-Charakter und föderaler Mitnutzung sind beispielsweise das Datenbankgrundbuch, das Vollstreckungsportal, die Insolvenzbekanntmachungen, das Maschinelle Mahnverfahren und das Zentrale Schutzschriftenregister. Das Mein-Justizpostfach (MJP) für natürliche Personen knüpft als Bürgerschnittstelle über die BundID an die Verwaltungsinfrastruktur an. Das Datenbankgrundbuch ist darüber hinaus Kandidat eines IT-PLR-Koordinierungsprogramms, das eine engere Abstimmung mit der föderalen Verwaltungsdigitalisierung adressieren könnte.

6.1.3 Übersicht der föderalen Basisdienste

Die folgende Tabelle listet die identifizierten föderalen Basisdienste alphabetisch auf und fasst je Dienst Beschreibung und API-Funktionen, eine Schlagwortliste der angebotenen Funktionen sowie typische Nutzergruppen zusammen.

Tabelle 50: Föderale Basisdienste

Basisdienst	Beschreibung
Basisregister für Unternehmen	Zentrales Register über Unternehmensbasisdaten, das Stammdaten und Identifikationsnummern aller in der deutschen Verwaltung geführten Unternehmen zusammenführt, qualitätsgesichert und zentral vorhält; Rechtsgrundlage Unternehmensbasisdatenregistergesetz (UBRegG, in Kraft seit 15. Juli 2021); registerführende Stelle ist das Statistische Bundesamt (am Standort Bonn); räumlich, organisatorisch und personell getrennt von den Bereichen der Bundesstatistik; Aufnahme des Betriebs 2025 Inhalt: Stammdaten (Name, Sitz, Geschäftsanschrift, Rechtsform, Wirtschaftszweig) sowie ausgewählte Identifikationsnummern aus weiteren Registern; Datenherkunft sind über 100 Quellregister mit Unternehmensbezug (z. B. Handelsregister, Vereins-, Genossenschaftsregister, Gewerberegister) Kernfunktion: Vergabe und Verwaltung der **bundeseinheitlichen Wirtschaftsnummer (beWiNr)** als register- und verwaltungsübergreifender Identifikator;

	technisch identisch mit der vom BZSt zugeteilten Wirtschafts-Identifikationsnummer (W-IdNr.) nach § 139c AO; dient im NOOTS-Architekturzielbild als IDM für Unternehmen API-Funktionen (im Aufbau, sukzessive Erweiterung des Nutzerkreises): • Stammdatenbereitstellung an angeschlossene öffentliche Stellen mit Rechtsgrundlage • Schnittstelle zum Organisationskonto des Portalverbundes für die Übernahme der Stammdaten in Online-Dienste Datenanlieferung der Quellregister an das Basisregister
Bezahldienste	Sammelkategorie föderaler Bezahlplattformen, die Online-Diensten und Fachverfahren die Abwicklung von Verwaltungsgebühren ermöglichen • **ZBDS (Zentrale Bezahldienste-Service):** vom IT-Planungsrat als Plattformkern-Bezahldienst des Deutschland-Stack benannt • **ePayBL:** in mehreren Ländern und Bundesbehörden eingesetzte Bezahlplattform; in ZBDS perspektivisch konsolidierend • **SAP Digital Pay und pmPayment:** weitere im Einsatz befindliche Plattformen einzelner Länder/Bundesbehörden API-Funktionen je Plattform: Anlegen einer Zahlungsanforderung mit Verfahrenskennzeichen, Beträgen und Verwendungszweck; Statusabfrage zum Zahlungsvorgang; Webhook-/Callback für Zahlungseingang; Erstattungs- und Stornoschnittstellen; Reconciliation-Schnittstellen für die Buchhaltung der Behörde; Anbindung an Payment-Service-Provider und an Haushalts-, Kassen- und Rechnungssysteme (HKR) • **XBezahldienste-Standard:** standardisiert nur die Kommunikationsbeziehung zwischen Online-Dienst und Bezahldienst für vorgelagerte Zahlungen (REST/OpenAPI 3, Beschluss IT-PLR 2023/51); deckt damit nur einen Teil der API-Funktionen der Bezahldienste ab. Periphere Schnittstellen zu HKR-Systemen, ZVPs sowie nachgelagerte Bezahlszenarien sind außerhalb des Standardisierungsumfangs und werden über plattform-spezifische Schnittstellen abgewickelt
BundID / DeutschlandID	Nutzerkonto Bund: Identifizierung natürlicher Personen für eGovernment-Verfahren, Login per SAML 2.0 mit verschiedenen Vertrauensniveaus (Selbstauskunft, ELSTER-Zertifikat, eID/Online-Ausweis, EU-notifizierte Mittel) entsprechend BSI TR-03107 Postfach- und Versanddienst: Empfang von Bescheiden und Versand von Nachrichten zwischen Bürgerpostfach und Behörden, REST-API seit März 2024 Statusmonitor: Abfrage des Verfahrensstands durch Bürger über REST-API
DVC-IAM	Identitäts- und Zugriffsmanagementkonzept der Deutschen Verwaltungscloud (DVC) für Behördenmitarbeitende, die in DVC-betriebenen Fachverfahren und Anwendungen arbeiten; selbst keine eigenständige IdP-Infrastruktur, sondern integrierender Rahmen für die jeweils bestehenden IdPs der nutzenden Behörden.

DVDV (Deutsches Verwaltungs-diensteverzeich-nis)	Zentrales Verzeichnis der elektronisch erreichbaren Behörden und Verwaltungsdienste in Deutschland; gibt für eine Verwaltungsleistung an einer Stelle die zuständige Behörde, ihre fachlichen Endpunkte sowie die zugehörigen Zertifikate und Routinginformationen zurück API-Funktionen: • Suche nach Diensten einer Organisation: liefert Verbindungsparameter wie Netzwerkadressen, Zertifikate, Schemata zu Inhaltsdaten, Anforderungen an das Signaturniveau • Suche nach Organisationen und Stellvertretern • Prüfung der Zugehörigkeit eines anfragenden Fachverfahrens zu einer Behördenkategorie
ELSTER	Gemeinsames eGovernment-Projekt aller Steuerverwaltungen von Bund und 16 Ländern auf Basis des Verwaltungsabkommens KONSENS, Federführung beim Bayerischen Landesamt für Steuern, technische Produktion und Service über die Zentrale Produktions- und Service-Stelle (ZPS Elster) in Nürnberg API-Funktionen: • Übermittlung strukturierter Steuerdaten (Einkommensteuer, Lohnsteuer, Umsatzsteuer, E-Bilanz, Anlage-spezifische Erklärungen u. a.) in den jeweils gültigen Datensatz-Schemata an die Steuerverwaltungen • Validierungs- und Plausibilitätsprüfungen vor Übermittlung • Empfang elektronischer Bescheide über ETR-Rückkanal in das ELSTER-Postfach • Authentifizierung über ELSTER-Anmeldemittel
EUDI-Wallet (natürliche Personen)	Staatlich getragene digitale Brieftasche für natürliche Personen auf Basis der eIDAS-2.0-Verordnung; Bündelung von Identitätsdaten und elektronischen Nachweisen auf dem Smartphone der Nutzenden, mit selektiver Offenlegung gegenüber Diensten API-Funktionen: • Identifizierung und Authentifizierung über Personenidentifizierungsdaten (PID) gegenüber öffentlichen und privaten Online-Diensten • Ausstellung und Speicherung elektronischer Attribut-Bescheinigungen (Electronic Attestations of Attributes, EAA), z. B. Führerschein, Hochschulzeugnis, ärztliche Bescheinigungen • Vorlage und Verifikation von PID und EAAs gegenüber Relying Parties mit selektiver Datenoffenlegung • qualifizierte elektronische Signatur direkt aus der Wallet • perspektivisch pseudonyme Logins und Zahlungsfunktionen Anbindung an die deutsche Verwaltung über zwei Wege vorgesehen: über die BundID oder direkt an die EUDI-Wallet Quelloffene staatliche Wallet-App; alternative Wallet-Anbieter können sich nach eIDAS-2.0-/ARF-Vorgaben zertifizieren lassen Stand: Sandbox-Erprobung seit Ende 2025, Go-Live der staatlichen EUDI-Wallet 02. Januar 2027
European Business Wallet	Geplantes EU-weites Pendant der EUDI-Wallet für juristische Personen, basierend auf einem Kommissionsvorschlag (2026); ergänzt die EUDI-Wallet für natürliche Personen um spezifisch unternehmensbezogene Funktionen

	Vorgesehene API-Funktionen: • Identifizierung von Unternehmen (Existenz, Rechtsform, Sitz, Vertretungsbefugnisse) gegenüber Behörden und Geschäftspartnern • Vorlage und Verifikation von Unternehmensnachweisen (Lieferkettennachweise, Compliance-Bescheinigungen, behördliche Genehmigungen) • Verwaltung qualifizierter elektronischer Siegel • Sicherer Kommunikationskanal zwischen Unternehmen und Behörden sowie zwischen Unternehmen • Unternehmensadressbuch über ein European Digital Directory Selbstständige und Einzelunternehmer können ihre EUDI-Wallet um Business-Wallet-Funktionen erweitern; rechtsverbindliches Handeln juristischer Personen erfolgt weiterhin über die EUDI-Wallets der vertretungsberechtigten natürlichen Personen Stand: Beratungen im Europäischen Parlament und Rat laufen, kein produktives System absehbar; Konzeption auf EU-Ebene, fernerer Umsetzungshorizont
FIM (Föderales Informationsmanagement)	Methode und Werkzeugbaukasten zur einheitlichen Beschreibung von Verwaltungsleistungen, Datenfeldern und Prozessen über alle föderalen Ebenen hinweg; Grundlage für Wiederverwendung in Onlinediensten und Fachverfahren Drei Bausteine mit jeweils eigenem XÖV-Standard: • **Leistungen** auf Basis von **XZuFi**: Austausch von Leistungssteckbriefen und Leistungsbeschreibungen (Stammtexten) zwischen Redaktionssystemen und nachnutzenden Systemen; Speicherung im Leistungskatalog (LeiKa) • **Datenfelder** auf Basis von **XDatenfelder**: Beschreibung von Datenfeldern, Datenfeldgruppen, Stammdatenschemata, Code- und Wertelisten als Grundlage für Antragsstrecken und Datenaustausch • **Prozesse** auf Basis von **XProzess**: standardisierte Prozessbeschreibungen für Verwaltungsabläufe, BPMN-2.0-basiert mit FIM-spezifischen Erweiterungen Zentrale Komponenten: • **FIM-Portal** als Web-Oberfläche zur Recherche und Information; bietet zugleich eine API für den Baustein Leistungen • **Sammelrepository** des Bausteins Datenfelder mit REST-API für Suche, Download und Upload von Schemata, Dokumentsteckbriefen und Codelisten sowie für die versionsstabile Auslieferung veröffentlichter Stände Pflege erfolgt dezentral: Bundesredaktion (Stammtexte zu Bundesrecht), Landesredaktionen (landesrechtliche Anpassungen), kommunale Redaktionen (Lokalinformationen)
FIT-Connect	Föderaler Antragsdatenübermittlungsdienst der FITKO; transportiert strukturierte Antragsdaten und Berichte (Einreichungen) von Onlinediensten zu den zuständigen Verwaltungssystemen, Ende-zu-Ende verschlüsselt API-Funktionen über drei APIs: • **Submission API:** Anlegen und Versenden von Einreichungen durch Onlinedienste, Auflisten und Abrufen abholbereiter Einreichungen durch Verwaltungssysteme, Zugriff auf das Event Log zur Statusverfolgung über die gesamte Einreichungslebensdauer

	• **Routing API:** Ermittlung des fachlich zuständigen Zustellpunkts auf Basis von Leistungs- und Ortsbezug der Einreichung, einschließlich der zugehörigen technischen Verbindungs- und Verschlüsselungsparameter • **Destination API:** Verwaltung von Zustellpunkten durch Verwaltungssysteme: Anlegen, Bearbeiten, Aktivieren und Stilllegen; Pflege der angebotenen Verwaltungsleistungen, Fachschemata, regionalen Zuständigkeiten, Rückkanaloptionen, Verschlüsselungsschlüssel und optional Listen erlaubter Sender
GDI-DE (Geodateninfrastruktur Deutschland)	Föderal aufgebaute Geodateninfrastruktur auf Basis der Verwaltungsvereinbarung GDI-DE und der INSPIRE-Richtlinie; setzt sich aus zentral betriebenen Komponenten und einer Vielzahl dezentraler Geodaten- und Geodatendienste-Anbieter in Bund, Ländern und Kommunen zusammen Vier Nationale Technische Komponenten, vom Bundesamt für Kartographie und Geodäsie (BKG) zentral betrieben: • **Geodatenkatalog.de:** Suchdienst, der Metadaten zu Geodaten und Geodatendiensten deutschlandweit über eine einheitliche Schnittstelle bereitstellt; bezieht die Metadaten harvestend aus den Katalogen des Bundes und der Länder über die OGC-Schnittstelle Catalogue Service for the Web (CSW) • **Geoportal.de:** Web-Plattform zur Recherche, Verknüpfung und Kartendarstellung von Geodaten der GDI-DE • **GDI-DE Registry:** Auskunftssystem für Namensräume, Codelisten und Koordinatenreferenzsysteme • **GDI-DE Testsuite:** Werkzeug zur Konformitätsprüfung von Geodaten und Geodatendiensten Dezentrale Geodatendienste der geodatenhaltenden Stellen auf Basis der OGC-Standards: WMS für Kartendarstellung, WFS für Vektordaten, WCS für Rasterdaten, CSW für Metadaten; in der Regel offen über das Internet zugänglich Autorisierungsbedarfe entstehen vor allem dort, wo Geodaten zugriffsbeschränkt sind: Liegenschaftskataster mit Personenbezug, lizenzpflichtige Geofachdaten, kostenpflichtige Premiumdienste
MUK (Mein Unternehmenskonto)	Bundesweit einheitliches Organisationskonto für Unternehmen, juristische Personen und vergleichbare Organisationen zur Identifizierung und Authentifizierung gegenüber Online-Diensten der öffentlichen Verwaltung sowie zum Empfang von Bescheiden über das Postfach 2.0; technisch auf Basis der ELSTER-Infrastruktur, Federführung Bayerisches Landesamt für Steuern, Bausteine 5 und 6 Bremen API-Funktionen: • **NEZO-Schnittstelle (Nutzung der ELSTER-Zertifikate im Rahmen des OZG):** SAML-basierte Identifizierungs- und Authentifizierungsschnittstelle für Online-Dienste; nach erfolgreicher Anmeldung Übermittlung von Stammdaten des Unternehmens und – bei Verknüpfung zu einem persönlichen ELSTER-Zertifikat – der handelnden Person • **Postfach 2.0:** zentrale Postfachfunktion für Unternehmen, mit Versand von Bescheiden und Mitteilungen durch Behörden sowie Empfang durch das Unternehmen; technische Anbindung der Behörden an Postfach 2.0 erfolgt

	über die lokal installierte Software ELSTER Transfer (ETR), inklusive M2M-Schnittstelle für große Datenmengen im Rückkanal • **Berechtigungssteuerung im MUK-Self-Service-Portal:** Verwaltung der Vorhaben (Online-Dienste, Portale, Fachverfahren) durch anbindende Stellen und Verwaltung der Bearbeitungsrechte für Mitarbeitende
NFK (Nationale Feedback-Komponente)	Mandantenfähige Lösung zur Erfassung anonymen Nutzendenfeedbacks zu Online-Diensten und Informationsangeboten der öffentlichen Verwaltung; vom BMDS bereitgestellt, vom ITZBund betrieben, seit 2021 im Regelbetrieb; erfüllt die Anforderungen aus Art. 25 SDG-VO (EU) 2018/1724 und übermittelt SDG-relevantes Feedback monatlich an die Europäische Kommission API-Funktionen: • **Eingebettetes Feedback-Formular:** in Online-Dienste oder Informationsseiten integrierbares Widget für direkten Feedback-Empfang durch die NFK; konfigurierbare Frage-Sets (einfaches und erweitertes Feedback) sowie eigene, nicht SDG-relevante Fragen • **REST-API zur Anlieferung von Feedback-Daten:** Stapel-Upload von Feedback-Einträgen aus eigenen Erhebungssystemen • **REST-API zur Abholung von Feedback-Daten:** Export der gesammelten Feedback-Einträge an angebundene Systeme
NOOTS (National Once-Only Technical System)	Föderale Vermittlungsinfrastruktur für den rechtskonformen Abruf elektronischer Nachweise aus den Registern der deutschen Verwaltung gemäß Once-Only-Prinzip; Staatsvertrag von Bund und Ländern Dezember 2024, iterative Einführung bis 2028, MVP seit Sommer 2025 in Erprobung, erste produktive Nachweisabrufe seit 2026; Federführung Bundesverwaltungsamt (BVA) Architektur folgt einer 4-Corner-Struktur mit Sicheren Anschlussknoten (SAK) bei Data Consumern und Data Providern und zentralen Vermittlungskomponenten dazwischen Zentrale Komponenten und ihre API-Funktionen: • **Registerdatennavigation (RDN):** liefert für einen angefragten Nachweistyp den zuständigen Data Provider und seine Verbindungsdaten in Form gesiegelter Zuständigkeits- und Verbindungstoken • **IDM für Personen (IDA-Verfahren):** Auflösung der Identifikationsnummer (IdNr) und der Basisdaten einer Person für den Nachweisabruf; betrieben durch das BVA, das die Daten aus der Steuer-ID-Datenbank des BZSt bezieht; XÖV-Standard XBasisdaten • **IDM für Unternehmen:** Auflösung der bundeseinheitlichen Wirtschaftsnummer (beWiNr) für den Nachweisabruf • **Vermittlungsstelle (VS):** führt die abstrakte Berechtigungsprüfung gemäß §§ 7 (2), 12 (4) IDNrG durch und protokolliert den Nachweisaustausch zur Verhinderung der Bildung von Persönlichkeitsprofilen • **Intermediäre Plattform:** Schnittstelle zum European-Once-Only-Technical-System (EU-OOTS) gemäß SDG-VO • **Datenschutzcockpit:** Auskunft an betroffene Personen über erfolgte Nachweisabrufe

	• **APIs der Data Provider:** dezentrale, von den jeweiligen registerführenden Stellen angebotene Schnittstellen zur Bereitstellung der Nachweise; Anbindung über den jeweiligen Sicheren Anschlussknoten an das NOOTS
OSiP (Online-Sicherheitsprüfung)	IT-Verfahren zur föderal koordinierten Durchführung personenbezogener Sicherheits- und Zuverlässigkeitsüberprüfungen; integriert Genehmigungsbehörden, Erkenntnisstellen (Landeskriminalämter, Verfassungsschutz) und Bundesregister in einen gemeinsamen, weitestgehend medienbruchfreien Workflow. API-Funktionen: • **Antragsannahme** durch den OSiP-Kern als Datendrehscheibe: Entgegennahme von Überprüfungsanträgen von anfragenden Stellen • **Erkenntnisanfrage:** regelbasierte Verteilung von Anfragen an die jeweils zu beteiligenden Erkenntnisstellen, Entgegennahme der Rückmeldungen und Weiterleitung an die anfragende Stelle • **Webservice-Schnittstellen auf SOAP/XML-Basis** für die direkte Anbindung von Fachverfahren der Genehmigungsbehörden, Erkenntnisstellen und Antragserfassungsstellen
OZG-RE (Onlinezugangsgesetz-konforme Rechnungseingangsplattform)	Zentrale Rechnungseingangsplattform des Bundes für den Empfang elektronischer Rechnungen (E-Rechnungen) im Standard XRechnung; betrieben durch die Bundesdruckerei im Auftrag des Bundes; seit September 2025 alleinige Eingangsplattform nach Konsolidierung der vorherigen ZRE; angeschlossen sind die unmittelbare Bundesverwaltung, Einrichtungen der mittelbaren Bundesverwaltung sowie kooperierende Bundesländer (Berlin, Brandenburg, Mecklenburg-Vorpommern, Sachsen, Thüringen) Übertragungskanäle für Rechnungssteller (Teil der API-Funktionen): • **Webformular** "Neue Rechnung erfassen" für die Direkterfassung im Portal • **Upload** einer fertig erstellten XRechnung-XML-Datei (max. 50 MB) • **E-Mail-Einreichung** mit verifizierter Absenderadresse • **Peppol-Webservice des Bundes:** kostenloser Maschine-zu-Maschine-Kanal über den Access Point der Bundesverwaltung; geeignet für Massenexport • **Status-API:** Einsicht in den Bearbeitungsstatus eingereichter Rechnungen (außer bei Einreichung über Peppol; dort über separate Verifizierung der Peppol-ID nachträglich nachvollziehbar)
PVOG (Portalverbund Online-Gateway)	Föderale Vermittlungsinfrastruktur für Verwaltungsleistungsdaten; aggregiert Beschreibungen, Metadaten und Zuständigkeitsinformationen aller deutschen Verwaltungsleistungen aus den Redaktionssystemen der 16 Landesredaktionen und der Bundesredaktion und stellt sie den Verwaltungsportalen sowie weiteren Abnehmern zur Verfügung; im FITKO-Portfolio, betrieben bei Dataport Datenstandard: **XZuFi** (Versionen 2.2 und 2.3.1) für die einheitliche Beschreibung von Verwaltungsdienstleistungen, Gebieten, Formularen und zuständigen Organisationseinheiten; tägliche Aggregation der Daten aus den angeschlossenen Redaktionssystemen API-Funktionen: • **Sammlerdienst:** Aufnahme und Aufbereitung von Verwaltungsleistungen aus den angeschlossenen Redaktionssystemen

	• **Bereitstelldienst:** zentrale Schnittstelle zum Abruf des Gesamtdatenbestands oder inkrementeller Änderungen im Format XZuFi 2.2 oder 2.3.1 • **Suchdienst:** Schnittstelle für die Suchmaschinen der Verbundportale, optimiert für Volltextsuche und Filter • **Suchclient:** Browser-basierter Zugang für Bürgerinnen, Bürger und Unternehmen
Statistischer Verbund Bund-Länder	Föderaler Verbund der Statistischen Ämter des Bundes und der Länder zur Durchführung der amtlichen Statistik in Deutschland; bietet zentrale Online-Dienste und APIs für Datenmeldung an die amtliche Statistik und Datenabruf veröffentlichter Statistiken API-Funktionen für die **Datenmeldung** durch auskunftspflichtige Stellen: • **Erhebungsportal:** zentraler Online-Zugang zu allen Online-Erhebungen der amtlichen Statistik • **IDEV (Internet Datenerhebung im Verbund):** Online-Formulare für die strukturierte Datenmeldung; Upload-Funktion für vorbereitete Datensätze; geeignet für kleinere bis mittlere Datenmengen • **eSTATISTIK.core (.CORE):** XML-basierte Schnittstelle (DatML/RAW) für die maschinelle Anlieferung großer Datenmengen aus Softwaresystemen der auskunftspflichtigen Stellen; Adressierung über die achtstellige Berichtseinheits-ID API-Funktion für den **Datenabruf**: • **GENESIS-Online-Webservice:** REST/SOAP-Schnittstelle für den automatisierten Abruf veröffentlichter amtlicher Statistiken aus der GENESIS-Datenbank des Statistischen Bundesamts; vergleichbare Webservices der Statistischen Landesämter (Regionaldatenbank Deutschland, Landesdatenbanken)
ZSK (Zentrale Statistik-Komponente)	Zentrale Plattform zur Erfassung und Auswertung nicht-personenbezogener Nutzungszahlen von Online-Diensten der öffentlichen Verwaltung; bereitgestellt vom BMDS (zuvor BMI), nach Pilotierung im ersten Halbjahr 2023 seitdem im Regelbetrieb; Teil der OZG-Rahmenarchitektur als zweite Basiskomponente neben der NFK Zentraler Indikator ist die monatliche, regional differenzierte Transaktionszahl eines Online-Dienstes API-Funktionen: • **Datenanlieferung:** REST-Schnittstelle für die Übermittlung der Transaktionszahlen je Online-Dienst • **Portalschnittstelle:** gesammelte Anlieferung der Transaktionszahlen aller in einem Portal befindlichen Online-Dienste in einem Aufruf • **Anlieferung über Webanalytik-Plattformen** wie Matomo • **Manuelle Anlieferung** über Weboberfläche per Formular oder CSV-Datei • **Auswertung** über die ZSK-Weboberfläche mit Filter nach Zeitraum, Portalen und Regionen
ZaPuK (Zielarchitektur Postfach- und Kommunikationslösungen)	Föderale Zielarchitektur und Umsetzungsprogramm für eine einheitliche Postfach- und Kommunikationsinfrastruktur der öffentlichen Verwaltung in Deutschland; Beschluss des IT-Planungsrats B-2025/28 in der 47. Sitzung als Baustein der Deutschland-Architektur; Federführung FITKO mit den Bundesländern Hamburg und Sachsen-Anhalt; im ersten Halbjahr 2026 läuft Phase 2 der

	Präzisierung und Validierung der Zielarchitektur und der Transitionswege, Phase 3 (technische Umsetzung) folgt Zielsetzung: Konsolidierung der heute heterogenen Postfachlandschaft (u. a. BundID/DeutschlandID, ELSTER-Postfach 2.0, Länderpostfächer, elektronischer Rechtsverkehr) zu einer gemeinsamen Infrastruktur; pro Nutzendengruppe (Privatpersonen, private Organisationen, öffentliche Stellen) ein zentraler Zugang Kernarchitekturmerkmale (gemäß Konzept V1.0): • **Bereitstellung von Postfach-Zugängen und Postfach-Backends auch durch Dritte:** nutzende öffentliche Stellen können eigene Postfachzugänge oder Backends betreiben (z. B. Verbindung mit E-Akten, Krisenredundanz, fachverfahrensspezifische Integration), unter Wahrung der Verschlüsselungsschicht (MLS, Ende-zu-Ende) • **Integration domänenspezifischer Lösungen:** Fachverfahren, Unternehmensanwendungen, Vorgangsbearbeitungssysteme können Nachrichten über die Infrastruktur senden und empfangen • **Kommunikationspartner:** Privatpersonen, private Organisationen und öffentliche Stellen; mindestens einer der Kommunikationspartner muss eine öffentliche Stelle sein Zentrale technische Anforderungen: Ende-zu-Ende-Verschlüsselung, Zero-Trust-Architektur, hohe digitale Souveränität für nutzende öffentliche Stellen API-Funktionen (im Aufbau, in Phase 2 zu präzisieren): • **Nachrichtenversand und -empfang** zwischen den Kommunikationspartnern • **Anbindung von Postfach-Zugängen** Dritter an die zentrale Infrastruktur • **Anbindung von Postfach-Backends** Dritter an die zentrale Infrastruktur

6.2 Mögliche Transitionsstrategien

Für die schrittweise Anbindung der bestehenden Basisdienste an die Föderale API-Autorisierungsinfrastruktur kommen verschiedene Strategien in Betracht, die jeweils unterschiedliche Aspekte priorisieren. Die folgenden vier Strategien werden im weiteren Vorgehen einzeln und in Kombination zu betrachten sein.

- **Erst mit den Willigen anfangen:** Eine Pull-Strategie setzt auf die Bereitschaft der Plattformverantwortlichen Stellen: Die Anbindung erfolgt zunächst dort, wo bereits eigenes Interesse, Modernisierungsdruck oder konkrete Anwendungsfälle bestehen, die von der zentralen Infrastruktur profitieren. Frühe Pilotumsetzungen entstehen so im Konsens und liefern Referenzen, die für spätere Anbindungen werben.
- **Nach Erneuerungszyklen:** Eine opportunistische Strategie bindet Basisdienste dann an, wenn ohnehin eine größere Erneuerung, Neuvergabe oder Major-Release-Migration ansteht. Die Anpassungen an die neue Infrastruktur können in den ohnehin laufenden Aufwand integriert werden, was wirtschaftlich vorteilhaft ist und Doppelarbeit vermeidet.

- **Nach strategischer Wichtigkeit:** Eine Priorisierungsstrategie nimmt zuerst diejenigen Basisdienste in den Blick, die als zentrale Querschnittsfunktionen besonders viele Anwendungsfälle und Nutzendengruppen betreffen, hohe Mengengerüste verarbeiten oder gesetzlichen Verpflichtungen unterliegen.
- **Nach Reifegrad:** Eine technikbezogene Strategie betrachtet den Ist-Stand der Basisdienste im Verhältnis zur Zielarchitektur. Dienste, die bereits OAuth-basierte Autorisierungsverfahren einsetzen oder eine moderne API-Architektur aufweisen, lassen sich mit deutlich geringerem Aufwand anbinden als Dienste mit eigenständigen, älteren Sicherheitsmechanismen.

Die vier Strategien sind keine Alternativen, sondern Achsen, die in der konkreten Roadmap zu kombinieren sind. Die konkrete strategische Auswahl und Planung kann jedoch erst im Rahmen eines Umsetzungsprojekts erfolgen, da es intensive Abstimmungen mit den verantwortlichen Stellen der jeweiligen Basisdienste erfordert.

7 Offene Fragen und Handlungsbedarfe

Die in den vorangegangenen Kapiteln dargestellte Zielarchitektur beschreibt den fachlichen, technischen und organisatorischen Rahmen einer föderalen API-Autorisierungsinfrastruktur. Ihre Realisierung wirft eine Reihe übergreifender Fragen auf, deren Klärung über den Geltungsbereich des vorliegenden Konzepts hinausgeht und die Befassung in den zuständigen Gremien sowie die Abstimmung mit angrenzenden Vorhaben voraussetzt. Das vorliegende Kapitel bündelt diese offenen Fragen und Handlungsbedarfe in thematischer Gliederung.

Die ersten vier Abschnitte behandeln strukturelle und übergreifende Themen: die gemeinsame Governance-Struktur für Basisdienste, das Zusammenspiel mit den Identity Providern von Behörden und sonstigen juristischen Personen, das Verhältnis zu netzseitigen Sicherheitsarchitekturen sowie die PKI-Infrastruktur. Die letzten beiden Abschnitte fassen die im Rahmen der Architekturbeschreibung in den Kapiteln 5.2 und 5.3 identifizierten Handlungsbedarfe zum übergreifenden Monitoring und zur revisionssicheren Protokollierung zusammen.

7.1 Gemeinsame Governance-Struktur für Basisdienste

Die Realisierung der föderalen API-Autorisierungsinfrastruktur und ihre dauerhafte Wirksamkeit setzen eine Governance voraus, die über die klassische Betriebsverantwortung einzelner Basisdienste hinausgeht. Mehrere Themen lassen sich ausschließlich auf Architektur- und Querschnittsebene wirksam koordinieren. Ohne entsprechende Strukturen ist eine erfolgreiche Realisierung unwahrscheinlich oder mit erheblichen Risiken behaftet.

Governance-Bedarfe der Zielarchitektur

Lifecycle- und Änderungsmanagement der Basisdienste: Technologische Weiterentwicklungen — etwa neue Versionen von Authorization-Server-Standards, Aktualisierungen kryptographischer Vorgaben oder direkte technische Anpassungen an einzelnen Basisdiensten — müssen über Akteurs- und Domänengrenzen hinweg koordiniert werden, damit Änderungen kompatibel, mit angemessenem Vorlauf und ohne Bruch in der föderalen Anschlussfähigkeit umgesetzt werden können.

Policy-Governance: Für das in Kapitel 5 beschriebene Berechtigungsmodell sind einheitliche Vorgaben und Prozesse zur Redaktion, Abnahme, Veröffentlichung und laufenden Überwachung von Policies erforderlich. Ohne solche Prozesse drohen inkonsistente Berechtigungsregeln und unklare Verantwortungszuschnitte zwischen Plattform- und Fachebene.

Attribut- und Identitäts-Governance: Die Wirksamkeit eines attributbasierten Berechtigungsmodells hängt von der Qualität, Konsistenz und Vergleichbarkeit der Attributinformationen ab. Erforderlich sind einheitliche Attributdefinitionen — für stamm- und leistungsbezogene Attribute bietet das Föderale Informationsmanagement (FIM) mit dem Baustein Datenfelder hierfür eine etablierte Grundlage — sowie eine Governance der beteiligten Identity Provider und Attribute Authorities hinsichtlich ihrer lokalen Verfahren, ihrer technischen Ausgestaltung und ihrer Schnittstellen.

Übergreifende Architekturgovernance: Die genannten Themen lassen sich nicht isoliert steuern. Eine übergreifende Architekturgovernance ist erforderlich, um sie aufeinander abzustimmen, Migrationspfade über mehrere Komponenten und Verantwortungsbereiche hinweg zu planen und konkrete Entwicklungsschritte mit den betroffenen Stellen zu verzahnen.

Verhältnis zu bestehenden Governance-Strukturen

Die Zielarchitektur bewegt sich in einem etablierten und sich derzeit weiterentwickelnden Geflecht föderaler Governance-Strukturen. Eine eigenständige, eigene Governance-Struktur für die föderale API-Autorisierungsinfrastruktur ist daher weder anzustreben noch realistisch erreichbar. Stattdessen ist eine Anbindung an die bestehenden und im Aufbau befindlichen Strukturen erforderlich, um Parallelstrukturen zu vermeiden und konsistente Entscheidungspfade zu gewährleisten. Relevant sind insbesondere:

- **IT-Planungsrat:** Das politisch-strategische Steuerungsgremium der föderalen IT, in dessen Zuständigkeit Beschlüsse zu Sicherheitsvorgaben, Standards und übergreifenden Architekturthemen fallen.
- **Föderales IT-Architekturboard:** Mit Beschluss 2024/26 hat der IT-Planungsrat das Rahmenkonzept „Föderales IT-Architekturmanagement" in der Version 2.0 verabschiedet. Das Architekturboard übernimmt die strategische Steuerung der föderalen IT-Architektur und ist für Architekturthemen mit übergreifender Relevanz zuständig.
- **Föderales IT-Standardisierungsboard:** Verantwortlich für die Festlegung und kontinuierliche Weiterentwicklung föderaler IT-Standards. Die in dieser Zielarchitektur referenzierten Sicherheitsvorgaben und Schnittstellenstandards sind in den Prozessen dieses Boards zu verankern.

- **DVC und DVC-Architekturboard:** Tragen die Vorgaben zu Verbindungsnetzen und netzseitiger Absicherung. Die in Kapitel 7.4 adressierten Abstimmungsbedarfe zwischen applikations- und netzseitiger Sicherheit sind in diesem Kontext zu führen.
- **FIM (Föderales Informationsmanagement) und insbesondere der Baustein Datenfelder:** FIM ist die etablierte föderale Struktur für die einheitliche Beschreibung und Pflege von Stamminformationen für Verwaltungsleistungen. Der Baustein Datenfelder stellt von rechtsetzenden Stellen freigegebene Datenfelder mit Namen, Definition und Identifikationsschlüssel sowie Datenfeldgruppen über zentrale Repositorys bereit; der Datenaustausch erfolgt über den Standard XDatenfelder. Die Attribut-Governance der föderalen API-Autorisierungsinfrastruktur — insbesondere für stamm- und identitätsdatennahe Attribute — ist mit den Vorgaben und Prozessen von FIM zu verzahnen, um parallele oder inkonsistente Definitionen zu vermeiden. Wo Attribute über den Geltungsbereich von FIM hinausgehen, etwa für plattformbezogene Attribute zu Software-Instanzen, sind ergänzende Strukturen erforderlich, die methodisch an FIM anschlussfähig sein sollten.
- **Produktmanagement-Modell des IT-Planungsrats:** Mit Beschluss 2024/51 vom 13. November 2024 hat der IT-Planungsrat das Produktmanagement-Modell in der Version 1.0 sowie die zugehörige Geschäftsordnung für die Gremien der Produkte und die Allgemeinen Teilnahmebedingungen beschlossen. Alle bestehenden und neuen Produkte des IT-Planungsrats — darunter FIM und Basisdienste wie FIT-Connect — werden durch die FITKO einheitlich nach diesem Modell gesteuert; jedes Produkt verfügt über ein eigenes Produktboard. Für die föderale API-Autorisierungsinfrastruktur ist zu klären, in welcher Form sie sich in dieses Modell einordnet, sei es als eigenes Produkt mit eigenem Produktboard oder durch eine geeignete Verzahnung mit den Produktboards der einbezogenen Basisdienste.
- **Deutschland-Architektur:** Mit Beschluss B-2026/04-IT vom 18. März 2026 hat der IT-Planungsrat eine erste Konkretisierung der strategischen Leitlinie der Deutschland-Architektur beschlossen, einschließlich eines Governance-Eckpunktepapiers. Die konkrete Governance-Struktur wird durch das FITKO-Architekturmanagement gemeinsam mit dem Bund unter Federführung der Freien Hansestadt Bremen erarbeitet und der 50. Sitzung des IT-Planungsrats zur Beschlussfassung vorgelegt. Die föderale API-Autorisierungsinfrastruktur ist von dieser Strukturbildung unmittelbar betroffen, da sie eine Querschnittsfunktion innerhalb der Deutschland-Architektur darstellt.

- **Deutschland-Stack:** Der mit Beschluss B-2026/03-IT vom 18. März 2026 gefasste einheitliche Plattformkern umfasst in seiner ersten Fassung die Basisdienste Identität und Vertrauen, Datenaustausch, Datenabruf, Zahlungsabwicklung und Postfach. Diese Domänen überlappen unmittelbar mit dem Geltungsbereich der föderalen API-Autorisierungsinfrastruktur, deren Mechanismen — Clientauthentifizierung, Berechtigungsentscheidung, Tokenausstellung — die Grundlage für eine koordinierte Nutzbarkeit dieser Plattformkern-Dienste bilden. Die Standards des Deutschland-Stacks werden gemäß Beschluss im Rahmen des Föderalen IT-Standardisierungsboards weiterentwickelt; insofern bestehen direkte Schnittstellen zwischen der Standardisierung der API-Autorisierungsinfrastruktur und der Standardisierung des Deutschland-Stacks.

Eine zentrale Herausforderung ergibt sich daraus, dass die föderale API-Autorisierungsinfrastruktur nahezu sämtliche der genannten Governance-Bereiche berührt, diese Bereiche jedoch in unterschiedlichen Kontexten und zu unterschiedlichen Zeitpunkten entstanden sind und sich bislang nicht durchgängig aufeinander abgestimmt entwickelt haben.

Die einzelnen Strukturen adressieren jeweils einen Teilaspekt — Architektur, Standardisierung, Produktsteuerung, Daten und Attribute, Netz- und Cloud-Vorgaben, strategische Plattformbildung — und verfügen über eigene Aufgaben, Beschlusslagen und Beteiligtenkreise. Insbesondere die jüngeren strategischen Initiativen Deutschland-Architektur und Deutschland-Stack sind dabei, eigene Governance-Strukturen zu etablieren, deren Verhältnis zu den bestehenden Strukturen — namentlich dem Föderalen IT-Architekturboard, dem Föderalen IT-Standardisierungsboard und dem Produktmanagement-Modell — selbst noch in Entwicklung ist.

Die föderale API-Autorisierungsinfrastruktur ist von dieser noch nicht abgeschlossenen Abstimmung in besonderer Weise betroffen, da sie als Querschnittsfunktion genau an der Schnittstelle dieser Strukturen positioniert ist. Eine wirksame Realisierung erfordert daher nicht nur die Anbindung an die einzelnen Strukturen, sondern auch eine aktive Befassung mit der Frage, wie diese Strukturen im Hinblick auf die Zielarchitektur untereinander abgestimmt werden können.

Offene Fragen und Handlungsbedarfe

Aus der Verzahnung der Governance-Bedarfe mit den bestehenden Strukturen ergeben sich folgende Punkte, die im weiteren Verlauf zu klären sind:

- **Verortung der Governance der API-Autorisierungsinfrastruktur:** In welcher Konstellation der genannten Gremien werden die Architekturentscheidungen, die Standardpflege und die Migrationsplanung der föderalen API-Autorisierungsinfrastruktur dauerhaft verankert?
- **Verortung im Produktmanagement-Modell:** Wie wird die föderale API-Autorisierungsinfrastruktur im Produktmanagement-Modell des IT-Planungsrats eingeordnet — als eigenständiges Produkt mit eigenem Produktboard, als Querschnittsfunktion über die Produktboards der einbezogenen Basisdienste hinweg, oder über ein anderes Steuerungsmodell? Welche Konsequenzen ergeben sich daraus für die Verzahnung mit den FIM-Strukturen für die Attribut-Governance?
- **Abstimmung zwischen den Governance-Strukturen:** Wie kann sichergestellt werden, dass die relevanten Governance-Strukturen — insbesondere das Föderale IT-Architekturboard, das Föderale IT-Standardisierungsboard, das Produktmanagement-Modell, FIM sowie die im Aufbau befindlichen Strukturen der Deutschland-Architektur und des Deutschland-Stacks — im Hinblick auf die Anforderungen einer föderalen API-Autorisierungsinfrastruktur konsistent zusammenwirken und nicht zu sich widersprechenden Vorgaben, Mehrfacherfassungen oder Steuerungslücken führen?
- **Schnittstelle zur Deutschland-Architektur:** Wie wird die Zielarchitektur in die im Aufbau befindliche Governance der Deutschland-Architektur eingebettet, insbesondere im Hinblick auf das Architekturboard und die Funktionsbausteine der Deutschland-Architektur?
- **Schnittstelle zum Deutschland-Stack:** Wie werden die Mechanismen der API-Autorisierungsinfrastruktur in die Standardisierung der Plattformkern-Dienste des Deutschland-Stacks eingebracht, sodass eine konsistente, anschlussfähige Lösung entsteht und keine Parallelentwicklung zwischen den Initiativen erfolgt?
- **Lifecycle-Prozess für die technischen Standards:** Wie wird die kontinuierliche Weiterentwicklung der hier referenzierten Sicherheitsvorgaben, Schnittstellen und Architekturvorgaben im Rahmen des Föderalen IT-Standardisierungsboards organisiert, einschließlich angemessener Übergangsfristen für die Basisdienste?

- **Policy-Governance-Prozesse:** Welche Stellen, Prozesse und Werkzeuge sind für Redaktion, Abnahme, Veröffentlichung und Überwachung der Policies des Berechtigungsmodells zu etablieren, und wie werden diese mit der Verantwortung der Plattformverantwortlichen und Fachverbundverantwortlichen Stellen verzahnt?
- **Attribut- und IdP-Governance:** Welche Stellen sind für die Pflege des Attributkatalogs verantwortlich, welche Anforderungen gelten an Identity Provider und Attribute Authorities hinsichtlich ihrer lokalen Verfahren und technischen Schnittstellen, und wie wird die Konsistenz zu bestehenden Standards (insbesondere FIM und Vorgaben der Registermodernisierung) sichergestellt?

7.2 Identity Provider von Behörden und sonstigen juristischen Personen

Die Wirksamkeit eines attributbasierten Berechtigungsmodells steht und fällt mit der Vertrauenswürdigkeit der Identitätsattribute, auf denen Berechtigungsentscheidungen beruhen. Das FöPD kann Clients anhand ihrer Software-Statement-Attribute unterscheiden – etwa nach Behördenzugehörigkeit, sachlicher Zuständigkeit oder organisatorischem Typ. Diese Attribute sind jedoch nur dann belastbar, wenn sie aus einer vertrauenswürdigen, geprüften Quelle stammen und nicht allein auf Selbstauskunft der antragstellenden Stelle beruhen.

Gegenwärtig fehlt im deutschen Verwaltungsraum eine flächendeckende, interoperable Infrastruktur für föderierte Identitätsdienste. Behörden und sonstige juristische Personen betreiben, soweit überhaupt vorhanden, häufig isolierte Identity Provider ohne standardisierte Schnittstellen und ohne Einbindung in ein übergreifendes Vertrauensnetzwerk. Dies erzwingt im FöPD aufwändige bilaterale Integrationsprozesse, schränkt die Automatisierbarkeit der Attributvalidierung ein und erhöht das Risiko von Fehlkonfigurationen und ungeprüften Identitätsbehauptungen. Das Berechtigungskonzept des FöPD adressiert diesen Sachverhalt als strukturellen Handlungsbedarf.

Positive Referenzmodelle: DFN-AAI und eduGAIN

Im Wissenschafts- und Hochschulbereich existieren seit Jahren funktionierende Föderationsmodelle, die belegen, dass eine dezentrale, institutsübergreifende SSO-Infrastruktur in der Praxis skaliert. Die **DFN-AAI** (Authentication and Authorisation Infrastructure des Deutschen Forschungsnetzes) verbindet über 900 Institutionen und ermöglicht deren Mitgliedern, sich mit institutionellen Identitäten an Diensten anderer Mitglieder zu authentisieren. Die Integration in das internationale **eduGAIN**-Netzwerk erweitert diesen Vertrauensraum auf tausende Institutionen weltweit.

Wesentliche Erfolgsfaktoren dieser Infrastrukturen sind:

- ein klar definierter, verbindlicher Attributkatalog (insbesondere `eduPerson`-Attribute),
- eine zentrale Metadaten-Registry als Vertrauensanker,
- eine gemeinsame Governance-Struktur mit klaren Aufnahme- und Betriebsanforderungen,
- sowie die konsequente Nutzung offener Standards (SAML 2.0, perspektivisch OIDC).

Diese Merkmale sind direkt auf den Verwaltungskontext übertragbar.

Bestandsaufnahme: S.A.F.E. als partielles Vorbild

Im Bereich der Justiz existiert mit **S.A.F.E.** (Secure Access to Federated E-Justice/E-Government) eine föderale SSO-Infrastruktur, die Landesjustizbehörden eine gemeinsame Authentifizierungsschicht bereitstellt. S.A.F.E. definiert einen Attributkatalog, der sowohl Organisationsattribute (z. B. Behörde, Organisationseinheit) als auch personenbezogene Attribute (z. B. Rolle, Funktionskennung) umfasst. Dieser Attributkatalog ist ein konzeptioneller Vorzug, der zeigt, dass die Verwaltung durchaus in der Lage ist, domänenübergreifende Attributstandards zu etablieren.

Gleichwohl weist S.A.F.E. strukturelle Schwächen auf, die einer Übernahme als Vorbild für das FöPD entgegenstehen: Die Infrastruktur basiert nicht auf OAuth 2.0 und unterstützt keine direkte Integration mit modernen API-Autorisierungsflüssen. Eine standardkonforme Federation im Sinne von OpenID Connect Federation oder SAML-Metadatenverbünden ist nicht vorgesehen. S.A.F.E. ist auf den Justizbereich beschränkt und nicht für eine sektorübergreifende Nutzung konzipiert. Diese Einschränkungen machen S.A.F.E. zu einem wertvollen Referenzpunkt für den Attributkatalog, nicht jedoch für die technische Architekturgrundlage einer FöPD-kompatiblen SSO-Federation. Dennoch ist die Integration von S.A.F.E. (resp. dessen Weiterentwicklung ‘SAmOA’ auf Basis von OAuth2) als förderierter Identity-Provider für die Justiz in das FöPD als erstrebenswert zu betrachten.

Vgl. Abschlussbericht: Untersuchung der Machbarkeit einer bundeseinheitlichen Justizcloud, BMJ 2025, Kapitel 3.4.6.1 Internes Identitäten- und Berechtigungsmanagement

https://www.bmjv.de/DE/themen/digitales/digitalisierung_justiz/digitalisierungsinitiative/_articles/justizcloud_artikel.html

ELSTER MuK (Mein Unternehmenskonto)

ELSTER MuK ermöglicht juristischen Personen eine digitale Authentifizierung gegenüber Behörden und wird als Komponente für Identity-Management (IdM) für behördliche Identitätsinfrastruktur diskutiert. Für den vorliegenden Kontext weist MuK jedoch einige strukturelle Schwächen auf:

- Es wurde für die Authentifizierung von Unternehmen gegenüber Steuerbehörden konzipiert, nicht für die Authentifizierung von Behörden als handelnde Organisationen in einer M2M-Infrastruktur.
- Fachliche und örtliche Zuständigkeiten von Behörden werden nicht abgebildet; ein Attributkatalog, der für Berechtigungsentscheidungen im FöPD nutzbar wäre, ist nicht vorhanden.
- MuK ist zudem nicht federation-fähig im Sinne offener Standards und erlaubt keine direkte Integration in OAuth-2.0-basierte Autorisierungsflüsse. Damit fehlen wesentliche Merkmale zur Steigerung der Sicherheit, Verlässlichkeit und Effizienz in der Authentifizierungsschicht.

Es eignet sich damit allenfalls als Übergangslösung für die Authentifizierung privatwirtschaftlicher Akteure, nicht jedoch als Grundlage einer skalierbaren Identitätsinfrastruktur für vorwiegend behördliche Strukturen und Nutzer; und deren Anforderungen.

Anforderungen an eine SSO-Federation im FöPD-Kontext

Die FöPD-Infrastruktur setzt nicht auf einen einzelnen zentralen Identity Provider, sondern auf ein dezentrales Ökosystem von IdPs – darunter bestehende Lösungen wie ELSTER oder MuK für spezifische Akteursgruppen.

Ziel sollte der Aufbau einer eigenständigen, behördlichen Federation sein, in die sowohl das FöPD und die angeschlossenen Basisdienste als Relying Parties als auch die jeweiligen IdPs als Vertrauensanker eingebettet werden. Eine solche Federation schafft den übergreifenden Vertrauensrahmen, innerhalb dessen Identitäten und Attribute über Organisationsgrenzen hinweg verlässlich ausgetauscht werden können. Die Dopplung von Identitätsinfrastruktur ist zu vermeiden; bestehende Verzeichnisdienste und Authentifizierungssysteme sollen als Basis genutzt und föderationskonform erschlossen werden.

Die Integration in ein solches Netzwerk schafft mehrfachen Mehrwert: Attributangaben zu Clients staatlicher Stellen werden durch den jeweiligen institutionellen Identity-Provider

kryptographisch abgesichert und können im FöPD ohne bilaterale Einzelprüfung (als organisatorische Freigabeprozesse) als technisch vertrauenswürdig behandelt werden.

Die Automatisierung der Attributvalidierung im Rahmen der Software-Statement-Registrierung wird erheblich vereinfacht. Gleichzeitig profitieren angeschlossene Institutionen von einer gemeinsamen Authentifizierungsinfrastruktur, die Redundanz abbaut und Pflege- sowie Betriebsaufwand verteilt.

Mindestanforderungen an eine Identity-Federation im FöPD-Kontext

Eine Identity-Federation die im FöPD als vertrauenswürdige Identitätsquellen anerkannt werden soll, sollte folgende Mindestanforderungen erfüllen:

Vertrauensbeziehung (Federation): Alle teilnehmenden IdPs sollten in das gemeinsame Föderationsnetzwerk eingebunden sein, das durch eine zentrale Metadaten-Registry und ein definiertes Governance-Rahmenwerk konstituiert wird. Bilaterale Vertrauensbeziehungen ohne übergreifende Governancestruktur erfüllen diese Anforderung nicht. Technisch bevorzugt werden OIDC Federation (OpenID Connect Federation 1.0) oder SAML 2.0 Metadaten-Federation. Das FöPD und die Basisdienste nehmen als Relying Parties an derselben Federation teil und beziehen Identitäts- und Attributaussagen vorwiegend aus federation-konformen Quellen.

Einheitlicher Attributkatalog: Alle teilnehmenden IdPs sollten einen verbindlichen, federation-weit normierten Satz von Identitätsattributen auf Organisations- und ggf. Personenebene bereitstellen. Hierzu zählen auch Pflichtattribute wie zum Beispiel: eindeutige Organisationskennung, Organisationstyp, zuständige Jurisdiktion sowie fachliche und örtliche Zuständigkeiten, soweit durch ein Zuständigkeitsregister bestätigt. Der Attributkatalog sollte verbindlich spezifiziert, versioniert und durch die Federation-Governance normiert sein, sodass Relying Parties wie das FöPD und die Basisdienste Attribute über IdP-Grenzen hinweg einheitlich interpretieren und auswerten können.

Governance: Die Aufnahme eines IdP in das Föderationsnetzwerk muss an nachweisbare organisatorische und technische Voraussetzungen geknüpft sein. Dazu gehören die Überprüfung der Identität und Rechtspersönlichkeit der betreibenden Institution, die Verpflichtung zur Einhaltung des Attributkatalogs sowie definierte Prozesse für Betrieb, Störungsmanagement und Maßnahmen bei Verstößen.

Handlungsbedarfe und Empfehlungen

Die Etablierung einer föderalen SSO-Infrastruktur für den Verwaltungsbereich ist primär eine organisatorische und politische Aufgabe. Die technischen Standards sind vorhanden; es fehlt bislang an einem übergreifenden Mandatierungs- und Koordinierungsrahmen.

Folgende Empfehlungen richten sich daher an geeignete übergreifende Gremien:

- Bund, Länder und Kommunen sollten ertüchtigt werden, institutionelle Identity Provider einzuführen und in eine gemeinsame FöPD-Federation nach dem Vorbild der DFN-AAI einzubringen.
- **Beitrittsanreize** sollten so gestaltet sein, dass ein flächendeckender Beitritt realistisch ist. Die Verpflichtung zur IdP-Integration sollte für bestimmte Kategorien öffentlicher Stellen (z. B. oberste Bundes- und Landesbehörden) verbindlich werden; für Kommunen und sonstige juristische Personen ist ein stufenweises Modell mit definierten Übergangsfristen zu bevorzugen.
- Ein normatives **Behördenverzeichnis** mit bestätigten fachlichen und örtlichen Zuständigkeiten sollte als organisatorische Grundlage für die IdP-Federation geschaffen werden; die Zuständigkeitsbestätigung durch Aufsichtsbehörden sollte Teil des Aufnahmeprozesses sein.
- Ein **Attributkatalog-Gremium** sollte den normierenden Attributkatalog pflegen, versionieren und fortschreiben. Es sollte paritätisch mit Vertretern von Bund, Ländern, Kommunen und betroffenen Stakeholdern besetzt sein und seine Beschlüsse im Rahmen des FöPD-Governance-Prozesses konsentieren.
- Eine **Federation-Betriebsstelle** sollte benannt werden, die die zentrale Metadaten-Registry betreibt, Aufnahmeanträge prüft und die Einhaltung der technischen und inhaltlichen Anforderungen überwacht. Diese Stelle kann beim FöPD oder einer beauftragten Einrichtung des Bundes (z. B. BSI, FITKO) angesiedelt sein.
- Das FöPD sollte die Integration föderierter Identity Provider als bevorzugten Modus für die Attributvalidierung bei der Software-Statement-Registrierung unterstützen und entsprechende technische Schnittstellen bereitstellen.
- Für Behörden ohne eigene IdP-Infrastruktur sollte ein mandantenfähiger Behörden-IdP als Übergangslösung bereitgestellt werden.

- Analoge Grundlagen für privatwirtschaftliche Akteure – auf Basis geeigneter Registrierungsgrundlagen wie Handels- oder Branchenregistern – sollten geprüft und entwickelt werden.

Schließlich ist eine **Übergangsstrategie** für Institutionen ohne eigene IdP-Infrastruktur zu entwickeln. Für diese Fälle können "Mein Unternehmenskonto (MUK)" und BundID als Fallback-Lösung bereitgestellt werden, der diesen Institutionen eine schnelle Anbindung ermöglicht, ohne eigene Infrastruktur voraussetzen zu müssen.

7.3 Verhältnis zu netzseitigen Sicherheitsarchitekturen

Die im vorliegenden Konzept formulierten Vorgaben beschreiben eine applikationsbasierte Absicherung der föderalen API-Kommunikation auf Basis von OAuth, FAPI 2.0 und ergänzenden Mechanismen wie Sender-Constrained Access Tokens. Die zugrunde liegenden Profile schließen Fragen der Netzarchitektur und der netzseitigen Absicherung ausdrücklich aus ihrem Geltungsbereich aus. Die hier definierten Sicherheitsmaßnahmen sind so konzipiert, dass sie auch ohne ergänzende netzseitige Absicherung ein dem jeweiligen Schutzbedarf angemessenes Sicherheitsniveau für API-Zugriffe gewährleisten.

In der Sicherheitsstandardlandschaft der öffentlichen Verwaltung lassen sich zwei Pole unterscheiden, die das Verhältnis von Netz- und Applikationssicherheit unterschiedlich gewichten. Auf der einen Seite stehen netzarchitekturzentrierte Vorgaben, insbesondere der IT-Grundschutz-Baustein NET.1.1 sowie die DVC-Detailstandards 01 (Verbindungsnetz und Übergänge) und 48 (Sicherheit von API-Schnittstellen auf Netzwerkebene). Sie verankern Vertrauen primär in der Netzarchitektur und prägen die etablierte Sicherheitsarchitektur des öffentlichen Sektors. Auf der anderen Seite steht der Cloud Computing Compliance Criteria Catalogue (C5) des BSI, dessen finale Fassung C5:2026 im April 2026 veröffentlicht wurde. C5 adressiert in 17 Anforderungsbereichen Netz-, Identitäts-, Kryptographie- und Betriebsthemen als gleichrangige Stränge eines Layered-Defense-Modells und akzeptiert konstitutionell, dass Cloud-Dienste über das öffentliche Internet erreichbar sind. Die Kommunikationssicherheit (COS) zielt dabei auf Transportverschlüsselung und Segmentierung innerhalb der Anbieterumgebung, nicht auf den Betrieb in einem geschlossenen Verwaltungsnetz.

C5:2026 baut die Anforderungen in den Bereichen Identitäts- und Berechtigungsmanagement, Kryptographie, Container Management und Confidential Computing aus. Moderne Sicherheitsmodelle wie Zero Trust werden als Orientierung für die Cloud-Architekturentwicklung benannt. Gleichzeitig bleiben detaillierte Anforderungen an die Netzwerksicherheit bestehen. Damit ordnet C5 Netz-, Identitäts-, Kryptographie- und Betriebskontrollen als gleichrangige

Stränge eines mehrschichtigen Sicherheitsmodells ein. Die in diesem Konzept verankerten Mechanismen (FAPI 2.0, Sender-Constrained Tokens, dezentrale Policy-Infrastruktur, kontinuierliche Verifikation über das Shared Signals Framework, manipulationssichere Protokollierung im Transparency Log) entsprechen den identitäts- und kryptographiebezogenen Strängen dieses Modells und sind damit anschlussfähig an C5:2026.

Gleichzeitig ist die aktuelle Realität in der öffentlichen Verwaltung zu berücksichtigen: Eine erhebliche Zahl bestehender und in Entwicklung befindlicher Anwendungen wird heute in Verwaltungsnetzen betrieben, weil diese ein zentraler Bestandteil der etablierten Sicherheitsarchitektur des öffentlichen Sektors sind. Für die Akzeptanz und Anschlussfähigkeit der föderalen API-Autorisierungsinfrastruktur ist es daher zumindest mittelfristig erforderlich, dass die Infrastruktur mit dieser Rahmenbedingung umgehen kann. Insbesondere müssen Bestandslösungen integrierbar bleiben, ohne dass Betreiber gezwungen sind, ihre Netzanbindung grundlegend umzustellen.

Mittelfristig sollte geprüft werden, in welchem Umfang die in diesem Konzept und in den Sicherheitsvorgaben definierten applikationsbasierten Sicherheitsmechanismen (gegebenenfalls in Kombination mit einer C5-konformen Bereitstellung der Cloud-Infrastruktur) ausreichen, um API-Zugriffe für definierte Schutzbedarfe und Anwendungsszenarien sicher auch über das öffentliche Internet zu ermöglichen. Eine entsprechende Bewertung sollte in Abstimmung mit BSI und DVC erfolgen und kann in eine Aktualisierung der einschlägigen Empfehlungen und Vorgaben dieser Stellen münden. Damit würde der Spielraum für Betreiber erweitert, schutzbedarfsangemessene Anbindungsmodelle zu wählen, statt netzseitige Absicherung pauschal vorauszusetzen.

Aus dem Verhältnis der applikationsbasierten Sicherheitsvorgaben dieses Vorhabens zu den bestehenden netz- und cloudseitigen Standards ergeben sich folgende offene Fragen und Handlungsbedarfe:

- **Bestandsintegration:** Wie kann sichergestellt werden, dass Basisdienste und API-Clients, die heute in Verwaltungsnetzen betrieben werden, ohne grundlegende Anpassungen ihrer Netzanbindung an die föderale API-Autorisierungsinfrastruktur angebunden werden können?
- **Schutzbedarfsbezogene Bewertung:** Für welche Schutzbedarfe und Szenarien sind die applikationsbasierten Sicherheitsmechanismen (gegebenenfalls in Kombination mit einer C5-konformen Bereitstellung) ausreichend, um einen sicheren Betrieb über das öffentliche

Internet zu ermöglichen, und in welchen Fällen bleibt eine ergänzende netzseitige Absicherung sinnvoll oder erforderlich?

- **Aktualisierung von Empfehlungen:** Wie und in welchem Zeithorizont können auf Grundlage einer solchen Bewertung die einschlägigen Vorgaben von BSI (insbesondere zum IT-Grundschutz) und DVC (insbesondere zu den Detailstandards 01 und 48) angepasst werden, sodass applikationsbasierte Absicherung als gleichwertige Option für definierte Szenarien anerkannt wird?
- **Konsistenz der kryptographischen Vorgaben:** Die in den Sicherheitsvorgaben dieses Vorhabens referenzierten BSI-Vorgaben (insbesondere BSI TR-02102-2) sind mit den entsprechenden Vorgaben der DVC-Detailstandards abzugleichen, um widersprüchliche Mindestanforderungen zu vermeiden.
- **Governance:** Es ist eine geeignete Form der Abstimmung zwischen den verantwortlichen Gremien (insbesondere BSI, DVC und der Trägerschaft des vorliegenden Konzepts) zu etablieren, sodass Weiterentwicklungen der jeweiligen Standards aufeinander abgestimmt erfolgen.

7.4 PKI-Infrastruktur

Die Verwaltungs-PKI (V-PKI) ist die etablierte föderale Vertrauensinfrastruktur des öffentlichen Sektors für Zertifikate von Organisationen, Organisationseinheiten und IT-Systemen. Sie ist damit ein potenziell zentraler kryptographischer Baustein für die Absicherung von Kommunikation und Daten im föderalen API-Ökosystem. In ihrer derzeitigen Ausgestaltung weist sie jedoch eine Reihe technischer Lücken auf, die in begleitenden Vorhaben und Architekturdiskussionen identifiziert und adressiert worden sind.

Aus Sicht der föderalen API-Autorisierungsinfrastruktur stehen dabei die technischen Modernisierungsbedarfe im Vordergrund: Die heutigen Beantragungs- und Austauschprozesse sind weitgehend manuell organisiert, Zertifikate werden mit langen Laufzeiten und in vorgegebenen Profilen ausgestellt, und es fehlen automatisierbare, in moderne Betriebsumgebungen einbettbare Schnittstellen für Beantragung, Erneuerung und Validierung. Demgegenüber sind Fragen, die in einzelnen Beiträgen begleitender Diskussionen in Richtung einer Erweiterung der PKI um Aufgaben des Identitäts- und Zugriffsmanagements oder um zertifikatsbasierte Berechtigungssteuerung weisen, konzeptionell von der vorliegenden Zielarchitektur klar getrennt: Identitäten und Authentifizierung werden im FöPD und in den föderierten Identity Providern abgebildet, Berechtigungen im Policy-System auf Basis des FöPD-Attributkatalogs

(Kapitel 5). Eine Verlagerung dieser Aufgaben in die PKI vermischt Lifecycles, die in der Zielarchitektur bewusst getrennt geführt werden, und ist daher nicht Bestandteil der hier formulierten Modernisierungsperspektive.

Bezüge der Zielarchitektur zur V-PKI

Die föderale API-Autorisierungsinfrastruktur hat in ihren Architekturentscheidungen die aktuelle Lage der V-PKI bewusst berücksichtigt:

- **ADR-002 (Client-Authentifizierung)** wählt `private_key_jwt` statt mTLS unter anderem mit der Begründung, dass die V-PKI auf absehbare Zeit nicht in der Lage sein wird, eine mTLS-Absicherung in dem hier benötigten Umfang zu unterstützen, und dass eine reine Implementierung auf Applikationsebene aufwändige Abstimmungen mit zentralen Netzwerkbetriebsabteilungen vermeidet.
- **ADR-003 (Sender-Constraining)** wählt DPoP statt mTLS-gebundener Tokens, da DPoP keinen Aufbau zusätzlicher PKI-Infrastruktur erfordert und Clients ihr Schlüsselmaterial selbst verwalten können.
- **ADR-004 (Vertrauenswürdigkeit von Public Keys)** verankert den Public Key direkt in der signierten Software Statement Assertion. Dies vermeidet die Notwendigkeit, den Public Key über separate Bezugswege wie JWKS-Endpunkte einzuholen, deren Vertrauenswürdigkeit anderweitig — etwa über zusätzliches Pinning oder ergänzen-de PKI-Strukturen — abgesichert werden müsste.

Diese Entscheidungen sind pragmatisch motiviert: Sie erlauben einen schnellen, skalierbaren Aufbau der föderalen API-Autorisierungsinfrastruktur, ohne dass deren Wirksamkeit von einer vorgängigen Modernisierung der V-PKI abhängt. Sie stellen jedoch keine grundsätzliche Abkehr von einer PKI-basierten Absicherung dar. Innerhalb der zentralen Infrastruktur kommt mTLS mit einer begrenzten internen PKI zum Einsatz (ADR-006), wo Lifecycle-Management beherrschbar ist.

Mehrwert einer technisch modernisierten V-PKI für die Zielarchitektur

Eine technisch modernisierte V-PKI mit automatisierbaren Beantragungs-, Erneuerungs- und Austauschprozessen würde der föderalen API-Autorisierungsinfrastruktur an mehreren Stellen unmittelbaren Mehrwert bringen. Im Mittelpunkt stehen dabei nicht zusätzliche fachliche Aufgaben der PKI, sondern die Behebung heutiger technischer Defizite im Lifecycle und in der Validierung von Zertifikaten.

Automatisierter Lifecycle für Zertifikate. Die heute überwiegend manuellen Prozesse für Beantragung, Ausstellung und Erneuerung von Zertifikaten sind in modernen Cloud-, Container- und API-Betriebsumgebungen nicht tragfähig. Eine modernisierte V-PKI mit standardbasierter Automatisierung (insbesondere nach dem Vorbild des ACME-Protokolls), Self-Service-Funktionen und kurzlebigen Zertifikaten würde den Betrieb in den Basisdiensten und der zentralen Infrastruktur erheblich vereinfachen. Sie würde zugleich die Sicherheit erhöhen, da die Lebensdauer sensiblen Schlüsselmaterials begrenzt und die manuelle Übermittlung privater Schlüssel (etwa als P12-Container) vermieden werden kann.

Erweiterte Nutzung von Zertifikaten in der Architektur. Über die mit den ADRs 002 bis 004 adressierten Mechanismen hinaus benötigen Basisdienste und nutzende Systeme an vielen Stellen Zertifikate — etwa für die mTLS-Absicherung der zentralen Infrastruktur (ADR-006), für Signatur und Verschlüsselung auf der Applikationsebene oder für die Validierung von Kommunikationspartnern. Eine modernisierte V-PKI würde die einheitliche, automatisierte Bereitstellung solcher Zertifikate ermöglichen, ohne dass jeder Betreiber eine eigene PKI aufbauen oder kommerzielle Lösungen einkaufen muss.

Robuste Validierung und Statusinformationen. Die heutige Validierungsinfrastruktur, insbesondere der OCSP-Betrieb, hat in der Vergangenheit wiederholt zu Verfügbarkeitsproblemen in produktiven Umgebungen geführt. Eine modernisierte V-PKI mit modernen Validierungsmechanismen (etwa CRLite oder vergleichbaren Ansätzen, ergänzend dazu OCSP Stapling oder hinreichend kurzlebige Zertifikate als alternative Kompensationsstrategie) und Monitoring-APIs für den Zertifikatsstatus (Ablaufwarnungen, Sperrinformationen) würde Betreibern die zuverlässige Einbindung der V-PKI in produktive Umgebungen ermöglichen.

Vereinfachtes Management der zentralen Zertifikate des FöPD. Auch das FöPD selbst benötigt für Signaturen — insbesondere von Software Statement Assertions und Attributbeständen — sowie für seine eigene Schnittstellenabsicherung Zertifikate. Eine moderne V-PKI mit

automatisierter Ausstellung und Erneuerung würde diese Zertifikatsverwaltung erheblich vereinfachen und absichern.

Integrierte Bestellprozesse durch Verzahnung mit dem FöPD. Eine Registration Authority auf Seiten der V-PKI könnte mit dem FöPD verzahnt werden, sodass Zertifikatsbestellungen medienbruchfrei und automatisiert auf Basis der bereits im FöPD validierten Identitäten und Identifikatoren erfolgen. Das FöPD selbst nimmt dabei keine Aufgaben einer Registration Authority wahr, da diese mit hohen regulatorischen Anforderungen verbunden sind, die in der Verantwortung der V-PKI verbleiben sollten. Eine solche Verzahnung würde Doppelvalidierungen derselben Organisationsidentitäten in unterschiedlichen Vorhaben vermeiden und die Bestellprozesse für Betreiber erheblich vereinfachen.

Offene Fragen und Handlungsbedarfe

Aus dem Verhältnis der föderalen API-Autorisierungsinfrastruktur zur V-PKI ergeben sich folgende offene Fragen:

- **Modernisierung der V-PKI:** In welchem Zeithorizont und unter welcher Federführung wird eine technische Modernisierung der V-PKI umgesetzt, insbesondere im Hinblick auf automatisierte Beantragungs-, Ausstellungs- und Erneuerungsprozesse, robuste Validierungsmechanismen sowie die Vorbereitung auf Post-Quanten-Kryptographie?
- **Nachzug von Zertifikatsanwendungen in der Zielarchitektur:** Welche Komponenten und Schnittstellen der Zielarchitektur sollen — sobald eine modernisierte V-PKI verfügbar ist — auf zertifikatsbasierte Mechanismen migriert oder durch solche ergänzt werden, und welche Übergangspfade sind dafür geeignet?
- **Verzahnung einer V-PKI-Registration-Authority mit dem FöPD:** Wie kann eine Registration Authority der V-PKI mit dem FöPD verzahnt werden, sodass Zertifikatsbestellungen automatisiert auf Basis der im FöPD validierten Identitäten und Identifikatoren erfolgen können, während die regulatorischen Aufgaben einer Registration Authority bei der V-PKI verbleiben?
- **Verhältnis von Certificate Transparency zum Transparency Log der Zielarchitektur:** Wie wird eine etwaige Certificate-Transparency-Lösung der modernisierten V-PKI mit dem in Kapitel 5.3 beschriebenen Transparency Log der föderalen API-Autorisierungsinfrastruktur abgestimmt, sodass keine Parallelinfrastrukturen entstehen?

7.5 Übergreifendes Monitoring und Risikobewertung

Kapitel 5.2 beschreibt die SSF-basierte Monitoring- und Risikobewertungsinfrastruktur als föderale Signalschicht für sicherheitsrelevante Ereignisse. Die Umsetzung dieser Infrastruktur erfordert die Klärung mehrerer offener Punkte, die über die im Kapitel 5.2 dargestellten Architekturentscheidungen hinausgehen und im Rahmen der Realisierung gesondert zu bearbeiten sind:

- **Föderales SSF-Eventprofil:** Spezifikation der föderalen API-spezifischen Eventtypen als formales SSF-Profildokument, einschließlich SET-Struktur, Subject-Identifier-Konventionen und Event-Semantik. Zu klären ist, ob dieses Profil als eigenständige OpenID-Profilspezifikation oder als technischer Standard durch den IT-Planungsrat verabschiedet wird.
- **Risk-Score-Modell:** Definition, wie ein Risikowert für eine registrierte Software bzw. einen API-Consumer-Client berechnet, aggregiert und kommuniziert wird, einschließlich der Schwellenwerte, ab denen ein entsprechendes Signal eine automatische Reaktion in den dezentralen Policy-Infrastrukturen auslöst.
- **Risikoregeln und Eskalationspfade:** Spezifikation der Korrelationsregeln und Schwellenwerte in der zentralen SSF-Monitoring-Infrastruktur sowie der automatischen Reaktionsketten, insbesondere die Abgrenzung zwischen dem direkten SSF-Kanal zum PDP (unmittelbare Einzelentscheidung) und dem zentralen Policy-Änderungsweg über den PIP (systemweite Wirkung).
- **Datenschutz und Protokollierungspflichten:** Klärung, welche Ereignisdaten über Basisdienst-Grenzen hinweg übertragen und aggregiert werden dürfen, insbesondere im Hinblick auf personenbezogene Daten und die DSGVO. Das SSF sieht Privacy-Controls auf Stream-Ebene vor, deren Konfiguration im föderalen Kontext zu spezifizieren ist.

7.6 Revisionssichere Protokollierung und Auditierung

Kapitel 5.3 beschreibt die Transparency-Log-Infrastruktur zur revisionssicheren Protokollierung sicherheitskritischer Ereignisse der zentralen Infrastruktur und legt Tessera als Implementierungsgrundlage fest. Auch hier bestehen Themen, die im Rahmen der Realisierung weiter zu klären sind:

- **Erprobung und Reife von Tessera für diesen Use Case:** Tessera ist als Go-Bibliothek für beliebige Log-Personalities konzipiert und technisch auf den vorliegenden Use Case anwendbar. Die bisherige Produktionserfahrung konzentriert sich jedoch auf Certificate Transparency und Sigstore-Ökosysteme mit sehr hohem Schreibvolumen und öffentlichem

Lesezugang, die sich strukturell vom vorliegenden Anwendungsfall unterscheiden. Für die föderale API-Autorisierungsinfrastruktur sind Schreibvolumen, Eintragungsformat, Zugangskontrolle zur lesenden API und der Betriebskontext wesentlich anders. Vor einer produktiven Umsetzung ist daher ein dedizierter Proof-of-Concept zu empfehlen, der den konkreten Use Case unter realistischen Bedingungen erprobt.

- **Eintragungsschema:** Definition des konkreten Eintragsformats für die föderale API-Autorisierungsinfrastruktur. Zu spezifizieren sind die Felder eines Logeintrags für jede Ereignisklasse (SSA-Ausstellung, Policy-Änderung, administrative Ereignisse, IdP-Föderierungsänderungen) sowie das Serialisierungsformat.
- **Datenschutz und Protokollierungspflichten:** Klärung, welche Datenfelder in Logeinträgen personenbezogene Daten enthalten können und wie der Konflikt zwischen der DSGVO (insbesondere dem Recht auf Löschung) und der append-only Natur des Transparency Logs aufzulösen ist. Mögliche Ansätze sind die Protokollierung von Pseudonymen oder Hashes anstelle direkter Identifikatoren oder eine Fokussierung auf Ereignisse ohne Personenbezug.
- **Witness-Infrastruktur:** Prüfung, ob und wie eine Witness-Infrastruktur zur weiteren Stärkung der Manipulationsresistenz eingesetzt werden soll. Witnesses sind unabhängige Parteien, die den Checkpoint des Logs gegenzeichnen und damit sicherstellen, dass der Log-Betreiber keinen abweichenden Log-Zustand gegenüber unterschiedlichen Beobachtern präsentieren kann. Das transparency-dev-Ökosystem stellt hierfür Referenzimplementierungen bereit; deren Einsatz im föderalen Kontext ist gesondert zu bewerten.
- **Aufbewahrungsfristen:** Definition der gesetzlichen und betrieblichen Aufbewahrungsfristen für Logeinträge unter Berücksichtigung der einschlägigen Vorgaben, insbesondere BSI-Grundschutz, Haushaltsrecht und DSGVO.

7.7 Erweiterungspotenziale für künftige Anwendungsfelder

Die in diesem Konzept beschriebene Zielarchitektur adressiert die als prioritär identifizierten Kernherausforderungen der föderalen Verwaltung im Bereich der API-Autorisierung: eine einheitliche, vertrauenswürdige und nachvollziehbare Grundlage für den Zugriff auf föderale Basisdienste. Der Scope wurde bewusst auf diese Kernherausforderungen fokussiert, um eine in absehbarer Zeit umsetzbare und betrieblich tragfähige Lösung zu ermöglichen. Erweiterungen auf weitere Anwendungsfelder sollen schrittweise und bedarfsorientiert erfolgen, sobald die Grundinfrastruktur etabliert ist und sich die jeweils einschlägigen Standards stabilisiert haben.

Die Architektur ist auf Erweiterbarkeit ausgelegt. Im Umfeld der föderalen Verwaltung sowie in der internationalen Standardisierungslandschaft zeichnen sich bereits heute Entwicklungen ab, die für künftige Erweiterungen relevant werden können. Eine Aufnahme dieser Themen in den Scope der vorliegenden Zielarchitektur war zum aktuellen Zeitpunkt nicht angezeigt, da die zugehörigen Standards und Anwendungsmuster noch in Entwicklung sind und ihre Tragfähigkeit für den Einsatz in der öffentlichen Verwaltung erst erprobt werden muss. Im Rahmen der Weiterentwicklung der Zielarchitektur sollten insbesondere die folgenden Themenfelder beobachtet werden:

- **Autorisierung für KI-Anwendungen und agentische Systeme:** Mit der zunehmenden Verbreitung KI-gestützter Anwendungen, und insbesondere agentischer Systeme, die im Auftrag einer Person oder Organisation autonome Handlungen ausführen, entstehen neue Anforderungen an Authentifizierung und Autorisierung. Mit dem Model Context Protocol (MCP) hat sich ein offener Standard für die Anbindung von KI-Modellen an externe Werkzeuge und Datenquellen etabliert, dessen Autorisierungsmodell explizit auf OAuth 2.1 aufsetzt und dessen Adoption in produktiven Umgebungen rasch zunimmt. Ergänzend dazu hat die OpenID Foundation im Oktober 2025 mit der „Artificial Intelligence Identity Management Community Group" (AIIM-CG) und dem zugehörigen Whitepaper „Identity Management for Agentic AI" eine fundierte Bestandsaufnahme der einschlägigen Anforderungen vorgelegt. Daran anknüpfende Spezifikationsarbeiten bei OpenID Foundation und IETF (etwa Vorschläge wie OpenID Connect for Agents) adressieren die nachvollziehbare Repräsentation des Auftraggebers im Token, die Delegation von Befugnissen entlang von Agentenketten sowie die granulare Einschränkung agentischer Handlungen. Welche dieser Entwicklungen für die föderale Verwaltung relevant werden und wie sie in die Zielarchitektur eingebettet werden können, ist zu einem späteren Zeitpunkt zu prüfen.
- **Nutzerzentrierte Datenhaltung und dezentrale Berechtigungsverwaltung:** Das SOLID-Protokoll verfolgt einen komplementären Ansatz: Daten verbleiben in nutzerseitig kontrollierten Datenspeichern, und Anwendungen erhalten Zugriff auf einzelne Datensätze auf Grundlage von Berechtigungen, die die betroffene Person selbst verwaltet. Die zugehörigen Mechanismen (etwa Web Access Control und Access Control Policies) wurden gezielt für die nutzerseitige Berechtigungsverwaltung im Datenzugriffskontext entwickelt und unterscheiden sich strukturell von der hier beschriebenen API-Berechtigungsarchitektur. Eine konkrete verwaltungsseitige Auseinandersetzung mit diesem Ansatz liegt mit dem Positionspapier „Digitale Souveränität mit Solid – für interoperable und dezentrale

Datenökosysteme in der Verwaltung“ der Stabsstelle Digitalisierung der Landeshauptstadt Kiel in Zusammenarbeit mit der HTWK Leipzig und dem KIT vor, das im Rahmen des Konsultationsprozesses zum Deutschland Stack eingebracht wurde. Insbesondere in Szenarien, in denen Bürgerinnen und Bürger ihre Daten dauerhaft selbst verfügen und Verwaltungsverfahren auf Wunsch der betroffenen Person auf diese Daten zugreifen, kommt eine Verzahnung mit der föderalen API-Autorisierungsinfrastruktur in Betracht. Eine solche Verzahnung setzt eine eigenständige konzeptionelle Betrachtung voraus.

Über die genannten Beispiele hinaus sind weitere Erweiterungspfade denkbar, etwa die Übertragung der Architekturmuster auf zusätzliche Anwendungsdomänen außerhalb der heute betrachteten Basisdienste oder die Verzahnung mit angrenzenden europäischen Initiativen. Eine Erweiterung der Zielarchitektur um diese oder weitere Themenfelder bedarf jeweils einer eigenständigen konzeptionellen Betrachtung sowie einer Abstimmung mit der zuständigen Governance. Sie sollte auf Grundlage einer konkreten Bedarfsanalyse erfolgen und dabei die Tragfähigkeit der bestehenden Mechanismen sowie deren mögliche Erweiterung gegenüber vollständig eigenständigen Lösungsansätzen bewerten.

8 High Level Umsetzungsroadmap

Die Roadmap beschreibt einen inkrementellen Weg von der technischen Erprobung zur produktiven Integration der föderalen API-Autorisierungsinfrastruktur. Es handelt sich explizit um eine High Level Roadmap, die den Weg zum Ausbau der Infrastruktur aufzeigt, ohne detaillierte zeitliche Vorgaben festzulegen. Zudem sind weitere Schritte nötig, die aktuell nicht geplant sind oder geplant werden können. Siehe hierzu auch Kapitel 7 für offene Punkte sowie Kapitel 6 für einen dedizierten Abschnitt zur Transition bestehender Basisdienste. Die Roadmap ist in sechs Phasen unterteilt. Die 50. Sitzung des IT-Planungsrat beschreibt den Startpunkt der Roadmap.

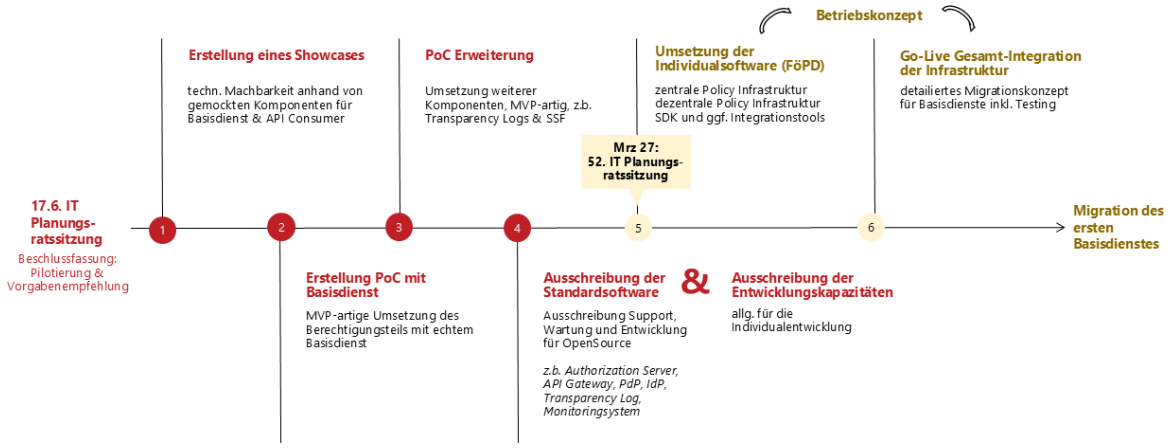


Abbildung 67: High Level Umsetzungsroadmap

In **Phase 1** steht die Validierung der technischen Machbarkeit der Zielarchitektur im Fokus. Ein "Showcase" validiert zentrale Architekturannahmen wie Policy Handling, Autorisierung und Schnittstellen möglichst früh im Projekt. Dabei kommen imaginäre Komponenten für Basisdienste und API Clients in einer Testumgebung zum Einsatz. Ziel ist es, frühzeitig Risiken zu erkennen und eine belastbare Grundlage für einen Proof of Concept (PoC) zu schaffen.

Phase 2 legt den Fokus weiterhin auf die Validierung von zuvor getroffenen Annahmen. Ein PoC soll basierend auf realen Basisdiensten in Testumgebungen die technische Machbarkeit weiter untermauern. Zentrale technische Aspekte sind die dynamische Clientregistrierung sowie die Autorisierung.

Die Ergebnisse der vorigen Phasen sollen in **Phase 3** kontinuierlich Richtung MVP weiterentwickelt werden, anhand dessen der Mehrwert der Infrastruktur klar ersichtlich wird. Die wichtigsten Funktionen sollen prototypisch implementiert sein oder in dieser Phase erstellt werden. Das Ziel der Phase ist neben der technischen Machbarkeit vor allem den Wertbeitrag der API-Autorisierungsinfrastruktur zu demonstrieren. Zentrale technische Aspekte sind die Erprobung des Transparency Logs sowie die Erprobung des Shared Signals Framework.

Phase 4 beschreibt die Beschaffung von Standardsoftware & Entwicklerkapazitäten für die folgenden Projektphasen. Insbesondere werden Ausschreibungen für Standardsoftware, Enterprise Support und Entwicklungsleistungen vorbereitet und durchgeführt. Ziel ist es, sowohl die Umsetzungs- als auch Betriebsfähigkeit sicherzustellen. Dabei werden bestehende Rahmenverträge geprüft und geeignete Liefermodelle definiert. Diese Phase stellt sicher, dass die Architektur nicht nur konzipiert, sondern auch nachhaltig betreibbar ist.

In **Phase 5** erfolgt die Realisierung der Individualsoftware (insbesondere FöPD). Zentrale und dezentrale Komponenten werden implementiert und über klar definierte Schnittstellen integriert. Darüber hinaus werden Integrationshilfen (bspw. SDKs) für Basisdienste erstellt, um die Adoption zu erleichtern.

Zum Abschluss werden in **Phase 6** die Komponenten schrittweise in den produktiven Betrieb überführt. Ein detailliertes Migrations- und Testkonzept stellt sicher, dass bestehende Basisdienste kontrolliert angebunden werden können.

Aufbauend auf der produktiven Infrastruktur können die ersten Basisdienste integriert werden, deren Auswahl im Laufe der Umsetzungsaktivitäten erfolgt und den Überlegungen folgt, die in Kapitel 6 skizziert wurden.

9 Anhang

9.1 Externe Referenzen auf Projektartefakte

Tabelle 51: Referenzen auf externe Projektartefakte

Projektartefakt	Beschreibung	Link
Anforderungen	Die Anforderungen wurden aus verschiedenen Quellen und von verschiedenen Systemen und Projekten eingesammelt, unter anderem DVC, FIT-Connect, NOOTS und xBezahldienste.	Die konsolidierten Anforderungen sind auf OpenCode veröffentlicht: https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/boards/3567
Architekturentscheidungen (Architecture Decision Records, ADRs)	Architekturentscheidungen (Architecture Decision Records, ADRs) beschreiben wesentliche Entscheidungen zur Systemarchitektur, die geprüften Alternativen, die Entscheidungstreiber sowie die Konsequenzen der getroffenen Wahl.	Alle ADRs sind in OpenCode veröffentlicht: https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/tree/main/Architekturentscheidungen
Architekturprinzipien	Die projektspezifischen Architekturprinzipien bilden den verbindlichen Handlungsrahmen für alle Architektur- und Designentscheidungen im Rahmen der föderalen API-Autorisierungsinfrastruktur.	Die vollständigen Beschreibungen einschließlich Begründungen, Abhängigkeiten und Auswirkungen sind im Projektrepository unter https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/blob/main/Architekturprinzipien/README.md verfügbar.
Glossar	Das Glossar definiert die zentralen Begriffe und Konzepte, die im Kontext der föderalen API-Autorisierungsinfrastruktur verwendet werden, über alle Dokumente hinweg.	https://gitlab.opencode.de/sachsen-anhalt/mid/foederale-api-autorisierungsinfrastruktur/-/blob/main/Glossar.md
FöPD-Mockup	Mockup-Design der Kernkomponente: Föderale Plattform Directory (FöPD). Das FöPD unterstützt zentrale Kernprozesse der föderalen API-Autorisierung. Die hier aufgeführten Mockups deuten zentrale Interaktion und Designs im Plattform Directory an und stellen keine Vollständigkeit	https://chart-willow-37308320.figma.site

	in Sinne der gesamten föderalen API-Autorisierungsinfrastruktur dar. Im Zielbild sollen alle Prozesse aus Kapitel 4.4 „Prozessübersicht und Darstellung der IT-Prozessunterstützung" der Zielarchitektur der föderalen API-Autorisierungsinfrastruktur abgebildet werden.	

Abbildungsverzeichnis

Tabellenverzeichnis