API Architektur Bund - Kurzversion

Klassifizierung	nicht klassifiziert
Status	genehmigt zur Nutzung
Programmname	Digitalisierungsstrategie des Bundes Strategische Initiative 3 «Once Only Prinzip»
Initiativenleiter	Wüst Jürg, DTI
Version	1.0
Datum	17. Januar 2022
Auftraggeber	Digitale Transformation und IKT-Lenkung (DTI)
Autor/Autoren	Brüning Fabian, Glavitsch Michael, Detecon (Schweiz) AG
Beteiligte Fachgruppen	Arbeitsgruppe API Architektur Bund, Architekturboard Bund (ABB)
Genehmigende Stelle	Andreas Spichiger, Digitale Transformation und IKT Lenkung

Inhaltsverzeichnis

1 Einleitung

«Als moderne Verwaltung vereinfachen wir für unsere Partner den Zugang zu unseren Behördenleistungen, indem wir die Leistungen über beliebige elektronische Mittel nutzbar machen.»[1]

Die API Architektur Bund verfolgt das Ziel, den digitalen Zugang zu Behördenleistungen im Bundesumfeld im Kontext Maschine-zu-Maschine für Unternehmen, Verwaltung und Personen zu standardisieren und zu fördern. Die API Architektur Bund wurde unter der Leitung des Bereichs Digitale Transformation und IKT-Lenkung (DTI) zusammen mit Vertretern aus zahlreichen Departementen und Verwaltungseinheiten (VE) erstellt. Auf Grund der fachlichen Tiefe und des breiten Umfangs der API Architektur Bund wurde diese Kurzversion mit dem Ziel erstellt, die wichtigsten Inhalte zusammenzufassen und somit die Verbreitung und Nutzung der API Architektur Bund zu steigern. An relevanten Stellen in der Kurzversion wird auf das entsprechende Kapitel im Hauptdokument verwiesen.

1.1 Ausgangslage

Mit der Digitalisierungsstrategie des Bundes 2020 -2023 hat der Bundesrat die Bundesverwaltung beauftragt, digitale Behördenleistungen zur Verfügung zu stellen. Der digitale Zugang zu Behördenleistungen und -daten erfolgt einerseits über E-Government-Portale oder Apps Mensch-zu-Maschine (H2M) oder direkt Maschine-zu-Maschine (M2M) über elektronische Schnittstellen. M2M-Anbindungen sind Zugangskanäle für digitale Behördenleistungen, welche auch von E-Government-Portalen oder Apps (H2M) genutzt werden können. Neben zahleichen existierenden E-Government-Portalen kommen vermehrt Forderungen nach einer direkten M2M-Anbindung.

Die API Architektur Bund ist Teil der «Stossrichtung C: Kunden- und Dienstleistungsorientierung» der Digitalisierungsstrategie 2020-2023. Innerhalb dieser Stossrichtung ist sie innerhalb der Strategischen Initiative «Once-Only Prinzip» im Massnahmenbereich «C2: Portale und Schnittstellen bereitstellen» angesiedelt, die darauf abzielt, Behördenleistungen über Portale und/oder elektronische Schnittstellen bereitzustellen und durch deren gemeinsame Nutzung Leistungen schneller, kostengünstiger und mit geringeren Risiken erbringen zu können.

1.2 Einordnung API Architektur Bund

Die Zielgruppe der API Architektur Bund umfasst Unternehmens- und IT-Architekten, Fachpersonal, Management, sowie weitere Orientierungssuchende aus Business und IT innerhalb und ausserhalb der Bundesverwaltung, welche Vorhaben zur digitalen Nutzung von Behördenleistungen planen oder umsetzen. Somit soll der digitale Zugang zu (offenen) Behördenleistungen im Kontext Maschine-zu-Maschine für Verwaltungseinheiten aller föderalen Verwaltungsebenen, für Unternehmen und für natürlichen Personen gefördert werden.

Die API Architektur Bund bietet als Referenzarchitektur mit Empfehlungscharakter sowie dazugehörigen Architekturprinzipien einen Ordnungsrahmen und eine gemeinsame Sprache für den Entwurf von API Lösungsarchitekturen. Die API Architektur Bund macht Gestaltungsempfehlungen zu verschiedenen Themen. Auf technische Einschränkungen bei der Gestaltung von Lösungsarchitekturen wird weitgehend verzichtet.

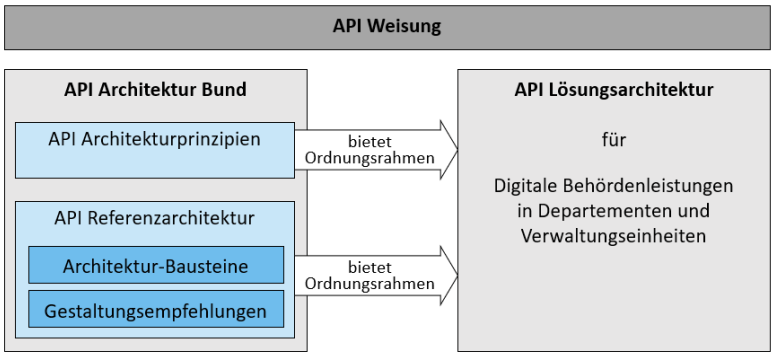

Abbildung 1: Positionierung der API Architektur Bund

Für die Weiterentwicklung der API Architektur Bund ist der Bereich DTI verantwortlich, welcher in Abstimmung mit den Departementen weitere Inhalte erarbeiten kann und ggf. Weisungen erstellt. Das Architekturboard Bund und das Gremium zu Datenmanagement/Dateninteroperabilität werden bei grösseren Änderungen konsultiert, während allfällige Weisungen dem Digitalisierungsrat zur Stellungnahme unterbreitet werden. Eine Anpassung soll mit Bedacht und langfristiger Perspektive erfolgen, damit die Stabilität für eine Zielverfolgung auf Architekturebene gewährleistet werden kann.

[1] Vision API Architektur Bund

2 Grundlagen & Geltungsbereich

In diesem Kapitel werden die wichtigsten Grundlagen und Begriffe definiert, welche in der API Architektur Bund einheitlich verwendet werden. Eine detailliertere Liste findet sich zudem in Kapitel 4 des Hauptdokumentes.

2.1 Grundbegriffe und Arten von Geschäftspartnern

Definition der wichtigsten Grundbegriffe:

Begriff	Beschreibung
Digitale Behörden-leistung	Der digitale Zugang zu einer Verwaltungsleistung, welche direkt via API oder indirekt über APIs via einem Behördenleistungsportal bezogen werden kann. Er kann auf einem einzigen API basieren, oder auf einem Verbund aus APIs aufbauen.
Endbenutzer	Eine natürliche Person, die IT Produkte bzw. Software nutzt, welche über APIs auf digitale Behördenleistungen zugreifen.
Lokale App	Eine lokal beim Endbenutzer laufende Applikation, welche auf eine digitale Behördenleistung zugreift. Dies umfasst Desktop Apps, Mobile Apps oder im Browser ausgeführter API-Zugriffscode sein (z.B. JavaScript).
Fachservice	Die Software-Komponente, welche die Fachlichkeit einer digitalen Behördenleistung abbildet und diese über eine elektronische Schnittstelle anbietet.
Fachservice-Cluster	Ein Verbund von identischen Instanzen eines Fachservices. Das Clustering dient der Steigerung der Performance und/oder der Ausfallsicherheit.
Portalanwendung	Eine mit einem Browser ansteuerbare Webapplikation. Eine Portalanwendung ist ein Spezialfall eines Fachservices und kann daher als API Client auftreten.
API Client	Die Software-Komponente, welche fähig ist, elektronische Schnittstellen anzusteuern und Antworten zu verarbeiten. Ein API Client kann entweder eine Lokale App, ein Fachservice oder eine Portalanwendung sein.
API Gateway	Die zwischen API Client und Fachservice vermittelnde Komponente. Die einfachste Ausführung eines API Gateways ist ein Reverse-Proxy mit zusätzlichen Fähigkeiten.

Tabelle 1: Grundbegriffe

Die von der Bundesverwaltung nach aussen angebotenen digitalen Behördenleistungen werden von verschiedenen Geschäftspartnern genutzt. Die folgenden Arten von Geschäftspartnern existieren.

Geschäftspartner	Beschreibung
Government	Internationale Verwaltungsebenen, Bundesverwaltungseinheiten (VE-übergreifend), Kantone und Gemeinden
Business	Juristische Personen: Unternehmen, Universitäten, Forschungsinstitute, NGOs oder Vereine
Citizen	Natürliche Personen, die digitale Behördenleistungen via Apps/Portanwendungen nutzen

Tabelle 2: Definition der Geschäftspartner

Daraus ergeben sich drei Arten von Geschäftsbeziehungen zwischen VE und Geschäftspartnern: G2G, G2B und G2C, wobei in der API Architektur Bund der generische Begriff des Geschäftspartners Government (G), Business (B) und Citizen (C) abdeckt. An Stelle der obigen Geschäftspartner können Intermediäre treten. Dabei handelt es um Leistungserbringer ausserhalb der Bundesverwaltung, welch digitale Behördenleistungen für die Geschäftspartner stellvertretend ansprechen und dazugehörige Dienstleistungen anbieten.

2.2 API Typen

In der API Architektur Bund wird zwischen drei Typen von APIs unterschieden:

API Typen	Beschreibung
Public API	Public APIs sind öffentlich und können von Geschäftspartnern ohne oder mit Registrierung genutzt werden. Die Nutzungsbedingungen (=Nutzungsvereinbarung) sind definiert, kommuniziert und müssen eingehalten werden. Bei Nutzung mit Registrierung ist eine Identität zu hinterlegen. Danach erfolgt der Zugriff mittels Credential und Secure Token. Bei Nutzung mit Registrierung wird bei erfolgreicher Authentifizierung jeder Zugriffsantrag akzeptiert. Selbstregistrierung ist die Regel.

Partner API	Partner APIs verlangen vom Geschäftspartner immer die Registrierung einer Identität und können nur von berechtigten Geschäftspartnern genutzt werden. Zugriffsanträge werden geprüft und können akzeptiert oder abgelehnt werden. Der Zugriff erfolgt mittels Credential und Secure Token. Für die Autorisierung ist ein Berechtigungssystem für die individuelle Steuerung der Zugriffe erforderlich. Für die Authentisierung muss ein dem Schutzbedarf der Daten entsprechendes Verfahren gewählt werden.
Private API	Private APIs werden nur innerhalb einer VE genutzt und sind von aussen nicht zugänglich.

Tabelle 3: Definition der API Typen

Die API Architektur Bund ist auf Public APIs und Partner APIs ausgerichtet. Die API Architektur Bund kann jedoch auch für Private APIs angewendet werden, insbesondere, wenn diese das Potenzial haben, zu Public APIs oder Partner APIs zu werden. Public und Partner APIs können jeweils alle Geschäftspartnertypen bedienen.

2.3 Integrationspfade & Anwendungsfälle

Für die Bereitstellung von digitalen Behördenleistungen werden Fachservices über Integrationspfade (IP) in API Clients sowie in andere Fachservices integriert. Abbildung 2 visualisiert die fünf wichtigsten Integrationspfade und deren Zusammenhang mit verschiedenen Geschäftspartnern.[2]

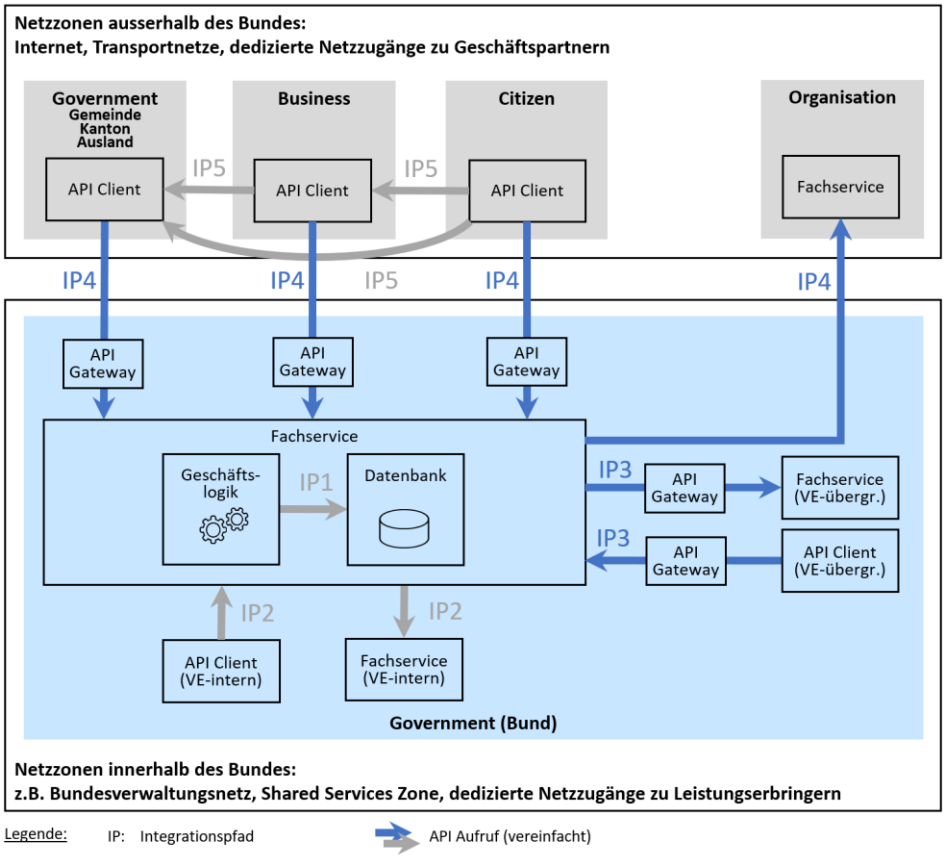

Abbildung 2: Integrationspfade

Die für die API Architektur Bund relevanten Integrationspfade und die dazugehörigen API Typen sind in Tabelle 4 beschrieben. Die API Architektur Bund befasst sich nur mit den Integrationspfaden IP3 und IP4. Bei IP3 handelt es sich um die VE-übergreifende Nutzung von digitalen Behördenleistungen innerhalb der Bundesverwaltung, während bei mit IP4 verbundenen API Aufrufen Bundes-Netzzonen-Grenzen überschritten werden. Bei diesen beiden Integrationspfaden sieht die API Architektur Bund als vermittelnde Systemkomponente zwischen API Client und Fachservice den Einsatz eines API Gateways vor. Bei IP4 kann zudem ein Intermediär an die Stelle des Geschäftspartners treten und damit eine Vermittler-Plattform die Rolle des API Clients wahrnehmen.

[2] Organisationen können gegenüber einer VE des Bundes eine digitale Leistung erbringen, welche wiederum in eine digitale Behördenleistung eingebunden wird. Diese Organisationen können einem Geschäftspartnertyp zugeordnet werden, sind hier aber von den Geschäftspartnern getrennt, da Geschäftspartner digitale Behördenleistungen beziehen, während Organisationen digitale Leistungen erbringen.

Pfad	Bezeichnung	Beschreibung	API Typen
IP1	FA-intern	Fachservice-interne Kommunikation	Private API
IP2	VE-intern	VE-interne Kommunikation zwischen zwei Fachservices oder zwischen einem API Client und einem Fachservice	Private API
IP3	Netzzonen innerhalb des Bundes	VE-übergreifende Kommunikation zwischen zwei Fachservices. Ein Fachservice kann als API Client auftreten.	Public API Partner API
IP4	Bundes-Netzzonen-Grenzen-überschreitend	Kommunikation zwischen lokaler App/Fachservice ausserhalb und Fachservice innerhalb der Bundesverwaltung	Public API Partner API
IP5	Netzzonen ausserhalb des Bundes	Kommunikation zwischen Anwendungen in Netzzonen ausserhalb der Bundesverwaltung	Public API Partner API

Tabelle 4: Integrationspfade

Eine digitale Behördenleistung kann erfordern, dass ein Fachservice ohne unmittelbar vorangehenden Request des Geschäftspartners ein API eines Geschäftspartners aufruft, was durch Integrationspfade IP3 und IP4 auch abgedeckt ist.

Für API Integration bei digitalen Behördenleistungen im Bundesumfeld treten sechs typische Anwendungsfälle auf (s. Abbildung 3 und Tabelle 5). Dabei wird zwischen Integration im Fachservice und Integration im API Client des Geschäftspartners unterschieden.

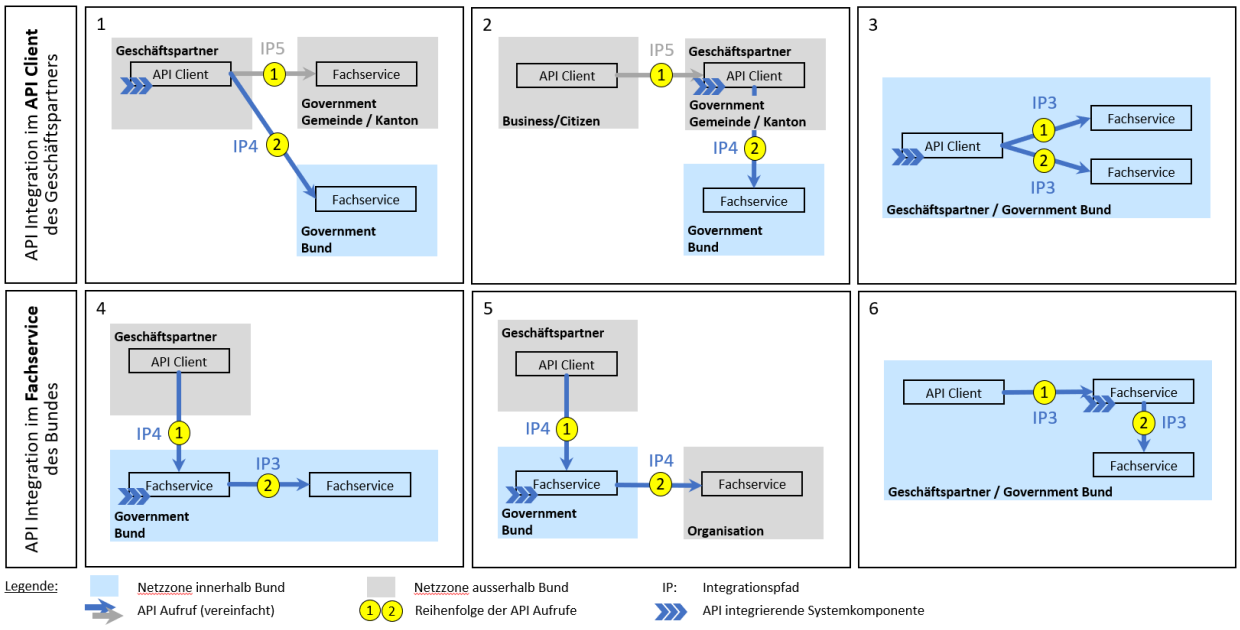

Abbildung 3: Anwendungsfälle für API Integration

API Integration im API Client des Geschäftspartners		API Integration im Fachservice des Bundes	
1	Ein Geschäftspartner ausserhalb des Bundes integriert 1-N API Aufrufe von Fachservices ausserhalb und innerhalb des Bundes im API Client.	4	Der Fachservice einer VE wird von einem Geschäftspartner ausserhalb des Bundes aufgerufen und integriert 1-N APIs anderer VEs des Bundes.
2	Ein Geschäftspartner ausserhalb des Bundes wird von einem Drittsystem aufgerufen und integriert für die Leistungserbringung gegenüber dem Drittsystem 1-N APIs von VEs des Bundes.	5	Der Fachservice einer VE wird von einem Geschäftspartner ausserhalb des Bundes aufgerufen und integriert 1-N APIs von Organisationen ausserhalb des Bundes.
3	Der Geschäftspartner ist eine VE des Bundes, dessen API Client 1-N APIs anderer VEs des Bundes integriert.	6	Der Fachservice einer VE wird von einem Geschäftspartner innerhalb des Bundes (andere VE) aufgerufen und integriert 1-N APIs anderer VEs.

Tabelle 5: Anwendungsfälle für API Integration

3 Architekturprinzipien

Die API Architektur Bund beinhaltet 10 Architekturprinzipien, welche als übergeordnete Prinzipien bei Bereitstellung von APIs und insbesondere bei der Gestaltung von API Lösungsarchitekturen dienen. Neben diesen gibt es übergreifende, für die gesamte Bundesverwaltung geltende (Architektur-)Prinzipien, z.B. die Tallinn Declaration on eGovernment[3] oder die SOA-Policies[4]. Unterhalb der 10 Architekturprinzipien der API Architektur Bund finden sich die konkreten Gestaltungsempfehlungen zu verschiedenen Themen.

Daneben existieren noch weitere API Guidelines, beispielsweise für REST- und ereignisorientierte APIs von Zalando[5] und den Schweizerischen Bundesbahnen[6]. Einzelne Architekturprinzipien der API Architektur Bund sind in denjenigen von Zalando und der SBB wiederzufinden, welche daher als detaillierte Ergänzung zu den nachfolgend formulierten Architekturprinzipien zu sehen sind.

Ref.	Name	Aussage
AP1	APIs haben Produktcharakter	APIs werden Geschäftspartnern wie Produkte mit zugehörigen Dienstleistungen angeboten. Jedes API hat einen bei der VE angesiedelten API Product Owner. Das API ist ein Teilprodukt des Gesamtpakets an Daten und Leistungen, welche das API zugänglich macht.
AP2	API First	Das funktionale Design von nach aussen gerichteten APIs wird als zentrales Element einer digitalen Behördenleistung definiert. Das Design richtet sich am Geschäftsfall der digitalen Behördenleistung sowie den benötigten Daten und Leistungen aus und wird in einem standardisierten Spezifikationsformat dokumentiert.
AP3	Schaffen von Transparenz zu verfügbaren APIs	Die API Metadaten und Anlaufstellen zu den in der Bundesverwaltung verfügbaren, nach aussen gerichteten APIs sind nach den Bestimmungen des EMBaG[7] in einem zentralen API Verzeichnis publiziert.
AP4	Einheitliche und vollständige Dokumentation von APIs	Die API Dokumentation von API Releases innerhalb der Bundesverwaltung ist einheitlich und vollständig.
AP5	Definieren und sicherstellen des Service Levels	Jedes API verfügt über definierte und transparente Service Level Objectives (SLO). Garantieren die API Dokumentation oder bei kostenpflichten APIs der Lizenzvertrag SLOs, so stellt der API Product Owner deren Einhaltung sicher.
AP6	Definieren der Daten- und Leistungsnutzung	Bei jedem API definiert eine Nutzungsvereinbarung, wie die dazugehörenden Daten und Leistungen genutzt werden dürfen. Das Nutzungsvereinbarung ist ein Bestandteil des Gesamtpakets an Daten und Leistungen, welche das API zugänglich macht.
AP7	Gewährleisten der Rückwärtskompatibilität	Bei der Weiterentwicklung von APIs wird Rückwärtskompatibilität sichergestellt.
AP8	Verwenden von anerkannten API Technologien	APIs in der Bundesverwaltung verwenden nur breit anerkannte API Technologien.
AP9	Abschirmen der API Clients vor Implementierungsdetails	Implementierungsdetails von Fachservices bleiben vor dem API Geschäftspartnern verborgen.
AP10	Gewährleisten der Idempotenz	Mehrfach aufeinander folgende identische API Aufrufe haben im Fachservice immer die gleiche Wirkung.

Tabelle 6: Architekturprinzipien der API Architektur Bund

Zwecks Umfang dieser Kurzversion werden Begründung und Auswirkung der einzelnen Prinzipien hier nicht aufgeführt. Diese sind, ebenso wie eine detaillierte hierarchische Einordnung der Prinzipien, in Kapitel 6 der Hauptversion der API Architektur Bund zu finden.

[3] https://www.admin.ch/gov/de/start/dokumentation/medienmitteilungen.msg-id-68342.html
[4] https://intranet.dti.bk.admin.ch/isb_kp/de/home/ikt-vorgaben/architekturen/r016-soa-policies.html
[5] https://opensource.zalando.com/restful-api-guidelines
[6] https://schweizerischebundesbahnen.github.io/api-principles
[7] https://www.admin.ch/gov/de/start/dokumentation/medienmitteilungen.msg-id-81580.html

4 Referenzarchitektur und Gestaltungsempfehlungen

In diesem Kapitel wird die API Referenzarchitektur mit Architekturbausteinen und Gestaltungsempfehlungen beschrieben. Die API Architektur Bund macht nur bei bestimmten Architekturbausteinen (=Architecture Building Blocks, ABBs) eine konkrete Gestaltungsempfehlung. Bei den anderen ist die Ausgestaltung den Leistungserbringern und VEs überlassen. Es sind die diejenigen ABBs definiert, welche für das Erbringen von digitalen Behördenleistungen sowie für die effiziente Interoperabilität als notwendig betrachtet werden.

Bei der Entwicklung einer API Lösungsarchitektur werden in Abhängigkeit vom Geschäftsfall nur ABBs der API Referenzarchitektur verwendet, welche zur Erfüllung der Geschäftsanforderungen einen Beitrag leisten. Für die Gestaltung der Lösungsbausteine bleiben weitgehende Freiheiten. Zwecks Nutzung von Synergien im Bundesumfeld wird der Einsatz von IKT Standarddiensten oder zentralen Anwendungen empfohlen.

4.1 Geschäftsfähigkeiten der Referenzarchitektur

Die API Architektur Bund definiert die Geschäftsfähigkeit API Management als Top-Level-Fähigkeit für API-spezifische Geschäftsfähigkeiten und weitere allgemeine IKT-Geschäftsfähigkeiten. Alle Geschäftsfähigkeiten werden hier den fünf sequenziellen API Prozessstufen **Manage API Lifecycle** (1), **Discover APIs** (2), **Register Client** (3), **Operate APIs** (4) und **Analyse API Operation & Usage** (5) zugeordnet. Die Geschäftsfähigkeit API Management wird umgesetzt, indem die API Lösungsarchitektur die API-spezifischen Geschäftsfähigkeiten umsetzt und die API-unspezifischen Geschäftsfähigkeiten einbindet.

Geschäftsfähigkeiten mit Gestaltungsempfehlung werden im entsprechenden Abschnitt erläutert. Andere Geschäftsfähigkeiten werden in Kapitel 7 der Hauptversion beschrieben. Ebenso finden sich dort Rollenbeschreibungen für API-spezifische Rollen, sowie das API Gouvernanzmodell der API Architektur Bund.

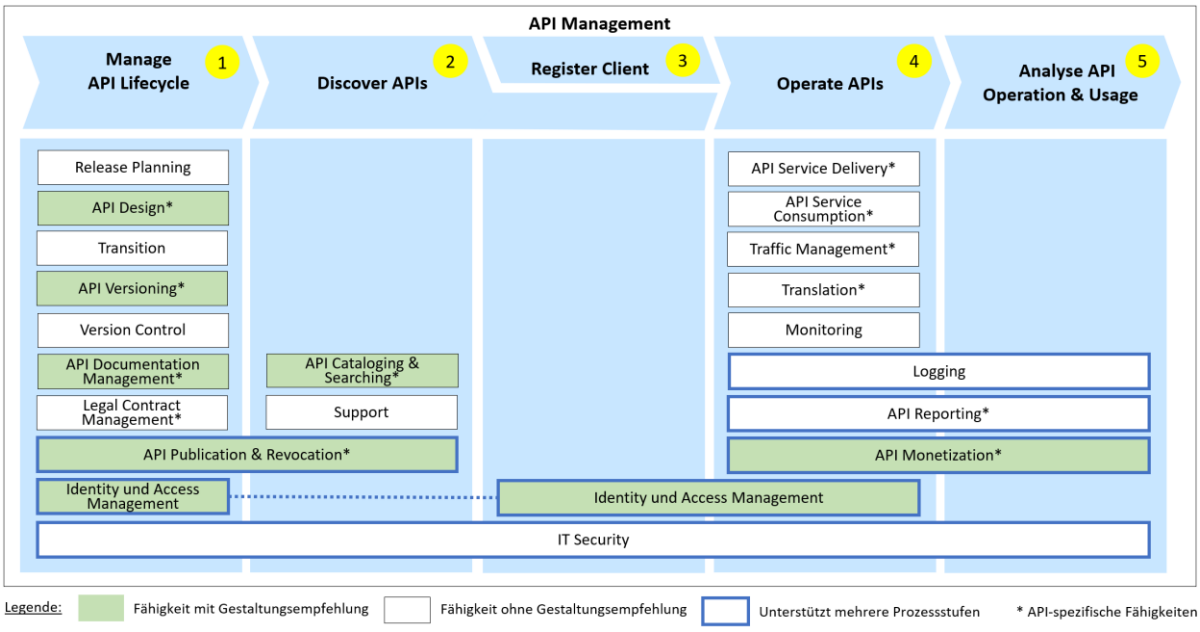

Abbildung 4: Landkarte der Geschäftsfähigkeiten

Die übergeordnete Prozessstufe **Manage API Lifecycle** beginnt mit der Planung der Entwicklung einer digitalen Behördenleistung inkl. API und endet nach der Stilllegung. Das API wird konzipiert, entwickelt, getestet, dokumentiert, publiziert, genutzt und schliesslich entfernt. API Lifecycle Management deckt alle API Versionen und Releases im Laufe des Lebenszyklus ab. Ein API Release ist eine API Version, die im API Verzeichnis publiziert wird. Die Publikation soll einen API Release für Unternehmens- und IT-Architekten, Fachpersonal, Management, sowie Orientierungssuchende aus Business und IT zur Verfügung stellen. Zudem werden API Metadaten zum API Release in einem API Verzeichnis eintragen.

Im API Verzeichnis sind veröffentlichte APIs publiziert, welche für die Entwicklung eigener APIs genutzt werden können. Die Fähigkeit API Cataloging & Searching ist somit zentral für die Prozessstufe **Discover APIs.** Bei APIs, die die Registrierung einer Identität voraussetzen, ist das API Verzeichnis der Startpunkt für die Prozessstufe Register Client.

Die Prozessstufe **Register Client** beinhaltet das Beantragen, Freigeben und Terminieren des Zugriffs auf ein API. Diese Prozessstufe wird übersprungen, wenn es sich um Public APIs ohne Registrierung handelt.

In der Prozessstufe **Operate APIs** erfolgt der Betrieb des APIs. Das API Gateway ist dabei das Kernelement des API, mit Vermittler-Rolle zwischen API Client und Fachservice. Es steuert als zentrales Eingangstor zur digitalen Behördenleistung die Zugriffe auf dahinterliegende Fachservices. Die Geschäftsfähigkeiten Logging und Monitoring sind dabei das Bindeglied zur Prozessstufe Analyse API Operation & Usage.

Die Prozessstufe **Analyse API Operation & Usage** umfasst das Messen von Kennzahlen zu APIs, deren Auswertung und das Erstellen von betrieblichen oder fachlichen Reports. Damit wird die Basis gelegt, um aus dem Betrieb von APIs Einnahmen zu generieren, insbesondere durch die Fähigkeiten API Reporting und API Monetization.

4.1.1 Gestaltungsempfehlung Transparenz

Die Gestaltungsempfehlung Transparenz beinhaltet Empfehlungen zu den nachfolgend beschriebenen drei Geschäftsfähigkeiten, welche nur in Kombination Mehrwert erzeugen und somit komplementär zu betrachten sind.

4.1.1.1 Empfehlung API Documentation Management

Die Fähigkeit API Documentation Management beinhaltet das Erstellen und Nachführen der Fachdokumentation inkl. der Nutzungsvereinbarung, der API Metadaten und der API Dokumentation im Rahmen des API Lifecycle Managements. Die API Architektur Bund empfiehlt die folgende Ausprägung dieser Fähigkeit:

- Die Artefakte der Dokumentation werden erstellt und fortlaufend gepflegt. Dazu gehören **API Metadaten** und **API Dokumentation**, für welche der API Product Owner verantwortlich ist, sowie die **Fachdokumentation**, für welche die Fachabteilung der dazugehörigen VE zuständig ist.
- Die **API Metadaten** stellen eine strukturierte, maschinell verarbeitbare, nicht versionierte Beschreibung des API Profils dar, um Informationen von APIs im Bundesumfeld zu veröffentlichen. Sie verweisen auf die versionierte **API Dokumentation**, welche relevante Informationen für einen Bezugsentscheid umfasst sowie Informationen für die technische Ansteuerung durch einen API Client. Insbesondere enthält die API Dokumentation auch die API Spezifikation, die API Release Notes und die SLOs, und verweist auf die **Fachdokumentation**. Diese ist Voraussetzung für die API Dokumentation, um zu beschreiben, welche Fachlichkeit ein bestimmter API Release abdeckt. Die Fachdokumentation beinhaltet die Nutzungsvereinbarung. Die Fachdokumentation und API Dokumentation werden in einer vom API Anbieter verwalteten, dezentral geführten **Landing Page** veröffentlicht.
- Die API Dokumentation wird so weit möglich automatisiert erstellt. Dazu werden anhand Schnittstellenbeschreibungssprachen (Interface Definition Language, IDL) API-Spezifikationen erstellt, aus denen automatisiert Dokumentationen, Server-/Client-Code-Vorlagen sowie Test Code und/oder Test Cases erstellt werden können. Etablierte Standards zur Spezifikation von RESTful APIs sind die OpenAPI Specification (OAS)[8] oder AsyncAPI[9]. Pro API Release sollte eine dokumentierte Version der API Dokumentation erstellt werden.

4.1.1.2 Empfehlung API Cataloging & Searching

API Cataloging & Searching ist die Fähigkeit, API Metadaten in standardisierter Form zu veröffentlichen[10] und die Bezugsmöglichkeit von APIs anzubieten. Die API Architektur Bund baut auf den Standards ISA[11] und eCH-0200[12] auf. Geschäftspartner und/oder beauftragte Leistungserbringer sollen mit Informationen versorgt werden, um Entscheide zum Bezug fällen zu können. Dabei wird bei API Metadaten in der API Architektur Bund zwischen der Datensicht und der Leistungssicht unterschieden. Beide Sichten sowie dazugehörige Standards sind in Kapitel 7.2 der Hauptversion der API Architektur Bund beschrieben.

Die API Architektur Bund sieht vor, ein **zentrales API Verzeichnis** im Bundesumfeld zu führen, in welchem Metadaten zu APIs von digitalen Behördenleistungen veröffentlicht werden. Daneben werden in vom API Verzeichnis referenzierten, **dezentral geführten Landing Pages**[13] mit API Dokumentationen die publizierten, versionierten API Releases beschrieben. Das API Verzeichnis ist öffentlich und enthält Metadaten zu solchen APIs, bei denen keine restriktiven Bedingungen zu den dahinterliegenden Daten gemäss EMBaG vorliegen. Im Fall von restriktiven Bedingungen empfiehlt die API Architektur Bund, den API Anbietern die Landing Page nur einem kontrollierten Nutzerkreis zugänglich zu machen. Bei Bedarf kann ein API Anbieter auch ein eigenes, dezentral geführtes API Verzeichnis betreiben, wobei die API Architektur Bund empfiehlt, dieses in das zentrale API Verzeichnis einzubinden. Im API Verzeichnis sollen weiterhin auch Informationen zu web-basierten Behördenleistungsportalen zu finden sein.

---

[8] https://swagger.io
[9] https://www.asyncapi.com
[10] Die Veröffentlichung von API Metadaten folgt den Bestimmungen des EMBaG - Bundesgesetzes über den Einsatz elektronischer Mittel zur Erfüllung von Behördenaufgaben: https://www.admin.ch/gov/de/start/dokumentation/medienmitteilungen.msg-id-81580.html
[11] ISA - Interoperability solutions for public administrations, businesses and citizens (ein Standard der Europäischen Kommission).
[12] https://ech.ch/index.php/de/standards/60609
[13] Auf Landing Pages kann verzichtet werden, falls Entwicklung und Betrieb nicht wirtschaftlich sind (z.B. Benutzerkreis zu klein).

Die API Architektur Bund empfiehlt darüber hinaus, API Metadaten von API Verzeichnissen auf einer Linked Data[14] Plattform zu veröffentlichen, womit die API Metadaten des API Verzeichnisses selbst über ein Public API abgerufen werden können.

4.1.1.3 Empfehlung API Publication & Revocation

Nach Entwicklung wird der API Release im API Verzeichnis publiziert. Die Publikation umfasst auch die Hinterlegung des API Builds und der API Konfiguration im Repository resp. der Registry. Am Ende des Lebenszyklus eines API Release wird dieser stillgelegt. Die Stilllegung (Revocation) erfordert wegen der Auswirkungen auf Geschäftspartner, Entwicklungs- und Betriebsressourcen eine sorgfältige Planung.

Die API Architektur Bund empfiehlt bei der Fähigkeit API Publication & Revocation, dass der API Product Owner als verantwortliche Person die folgenden Aufgaben wahrnimmt:

- Veröffentlichung der API Metadaten im API Verzeichnis und der zum API Release zugehörigen Online-Dokumentation auf der Landing Page, oder Delegation dieser Aufgabe an einen Publisher. Sicherstellung, dass alle berechtigten Personen Zugriff auf die Landing Page erhalten durch Rollenzuweisungen in eIAM.
- Publikation und Anpassung von API Metadaten bei einem API Release, falls diese bei publizierten API Releases Änderungen erfahren.
- Publikation und Anpassung der Online-Dokumentation auf der Landing Page.
- Sicherstellung des Deployment von API Instanzen auf der produktiven Umgebung. Abschluss von dazugehörigen Betriebsvereinbarungen (SLAs) mit den API Betreibern, Sicherstellung der Einhaltung der Service Level Objectives (SLOs).

4.1.2 Gestaltungsempfehlung API Monetization

API Monetization ist die Fähigkeit, aus dem Betrieb von APIs Einnahmen zu generieren und beinhaltet die Verbindung von API Konsum oder Nutzungsplänen mit Preismodellen. Dazu gehört auch das Durchsetzen von Quotas für Zugriffe bei Überschreitung von Bezugslimiten. API Monetization basiert auf KPIs und Berichten der Fähigkeit Reporting. Obwohl im Bundesumfeld offene Daten gebührenfrei verfügbar sind, beispielsweise bei Open Government Data[15], gibt die API Architektur Bund Empfehlungen bei der Übertragung von Kosten auf den API Geschäftspartner. Voraussetzung für eine Kostenübertragung ist immer eine gesetzliche Grundlage, wie z.B. die bundesrätliche Verordnung SR 510.602 über Geoinformation (Geoinformationsverordnung, GeoIV).

Die API Architektur Bund unterscheidet zwischen verschiedenen Monetarisierungs- und Preismodellen. Diese sind in Kapitel 7.3 der Hauptversion im Detail beschrieben. Das Übertragen von Kosten an den API Geschäftspartner stellt erhöhte Anforderungen, da eine bezahlte, vertraglich garantierte Leistung erbracht werden muss.

Die API Architektur Bund empfiehlt bei der Monetarisierung von APIs die Sicherstellung der folgenden Voraussetzungen, um die Akzeptanz der API Geschäftspartner zu gewährleisten:

- Ein **Geschäftspartner- und Anwendungsorientiertes API Design** ist zu gewährleisten. Insbesondere soll dieser durchgängige Prozess zur Abwicklung von digitalen Behördenleistungen unterstützen und eine hohe Benutzerfreundlichkeit durch einen hohen Automatisierungsgrad aufweisen.
- API Entwickler sollten für Entwicklungszwecke ohne Hindernisse **Zugang zu API Testversionen** erhalten.
- Klar **definierte und kommunizierte SLOs** existieren und werden in jedem Fall eingehalten.
- **Supportleistungen für die API Nutzung** sind über den gesamten Lebenszyklus eines APIs bereitzustellen. Auch müssen API Infrastrukturen bei Monetarisierung bestimmte Funktionen anbieten können, beispielsweise das Überwachen der API Nutzung.

4.1.3 Informationsmodell und Informationsobjekte

Die API Architektur Bund verwendet zur Beschreibung der Datenflüsse in der Referenzarchitektur ein Informationsmodell, welches auf den fünf Prozessstufen basiert. Dieses Informationsmodell beschreibt die Informationsobjekte (Akteure, Serviceobjekte und Datenobjekte) dieser API Referenzarchitektur und deren Beziehung zueinander. Eine detaillierte Beschreibung des Informationsmodell und der Informationsobjekte findet sich in den Kapiteln 8 mit weiteren Erklärungen in Kapitel 10 der Hauptversion der API Architektur Bund.[16]

---

[14] Unter Linked Data versteht man auf Basis des Resource Data Frameworks (RDF) strukturierte Daten, welche zur Verwendung in Software-Anwendungen veröffentlicht werden (M2M-Kommunikation). https://www.w3.org/TR/rdf11-primer

[15] https://www.bfs.admin.ch/bfs/de/home/dienstleistungen/ogd.html

[16] Im Kontext des Informationsmodells bzw. der Datenarchitektur macht die API Architektur Bund zwei Gestaltungsempfehlungen, (GE API Design und GE Identity und Access Management). Diese sind in der Kurzversion in Kapitel 4.2 zu finden.

Während im Informationsmodell bei den miteinander kommunizierenden Informationsobjekten die Begriffe Client, Gatekeeper Service und Ressource Service verwendet werden, sind es in der Systemlandschaft der Referenzarchitektur die Begriffe API Client, API Gateway und Fachservice.

4.2 Systemlandschaft der API Referenzarchitektur

Die Systemlandschaft der Referenzarchitektur umfasst nach Prozessstufen gruppierte Systemkomponenten gemäss Abbildung 5. Dabei unterscheidet die API Architektur Bund zwischen der **API Infrastruktur** (Services und Komponenten) und den **API Instanzen**, welche die API-spezifischen Systemkomponenten der digitalen Behördenleistung darstellen. Entwicklung und Betrieb der API Infrastruktur können von einer anderen Organisationseinheit wahrgenommen werden als Entwicklung und Betrieb der API Instanzen. Die API Architektur Bund empfiehlt jedoch grundsätzlich, Entwicklung und Betrieb der API Infrastruktur und der API Instanzen unter die gleiche technische Verantwortung zu stellen. API Instanzen exponieren die Fachlichkeit einer digitalen Behördenleistung, während die API Infrastruktur die Betriebsumgebung dafür bieten.

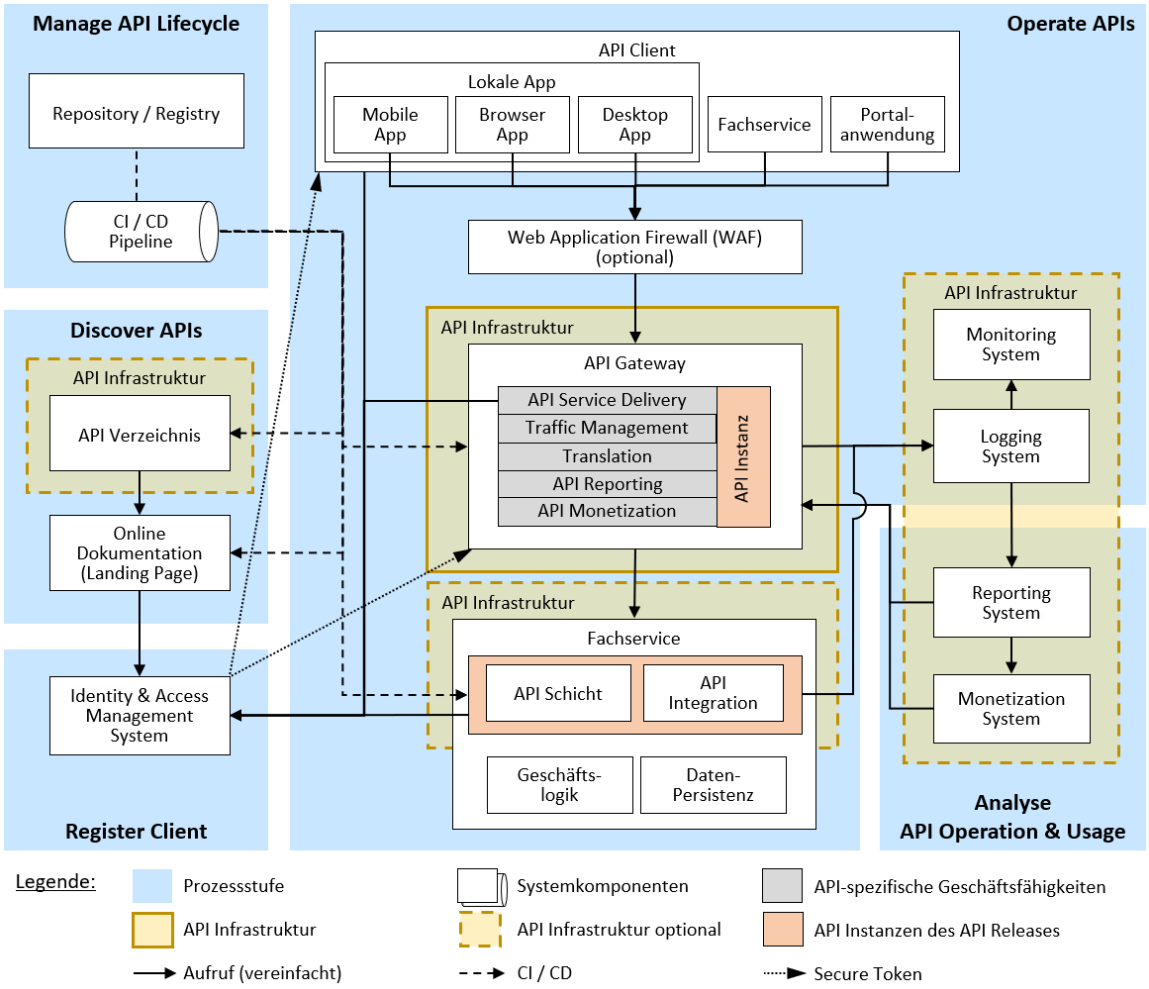

Abbildung 5: Systemlandschaft der API Referenzarchitektur

Die **API Infrastruktur** bietet die Systemkomponenten und Services für den Betrieb von APIs. Je nach gewähltem Produkt und Anwendungsfall kann die API Infrastruktur-Plattform unterschiedlich viele Systemkomponenten abdecken, was in Abbildung 5 durch die optionalen Komponenten veranschaulicht ist. Relevante Systemkomponenten müssen, falls sie nicht durch ein API-Infrastruktur-Produkt abgedeckt sind, durch die IKT Infrastruktur der VE ergänzt werden (z.B. ein Logging Service). Die API Infrastruktur kann auf mehrere geografische Standorte verteilt sein und damit eine API Instanz zwecks Traffic Management auch an mehreren geografischen Standorten unter verschiedenen Adressen anbieten. Die API Infrastruktur erlaubt die zentrale Verwaltung aller darin laufenden zum selben API gehörenden API Instanzen.

API Instanzen kommen an zwei Stellen zum Einsatz. Entweder laufen sie im API Gateway und konfigurieren dieses als Teil der API Infrastruktur bzw. stellen eigenständige ausführbare Objekte dar. Oder sie laufen im Fachservice in Form einer API Schicht oder in Form einer Systemkomponente für API Integration, welche ihrerseits als API Client auftritt und weitere APIs ansteuert. Das Repository resp. die Registry liefert die API Builds und API Konfigurationen für beide Arten von API Instanzen. Die Geschäftslogik der Fachwendung ist nie Teil der API Instanz, für

die Implementierung der Fachlichkeit ist die Geschäftslogik zuständig. Zwecks Sicherstellung der Skalierung können API Instanzen vervielfacht und an mehreren geografisch verteilten Standorten laufen. In reifen Umgebungen kann das Reporting System zwecks dynamischer Skalierung auch Steuerungsfunktionen unterstützen.

4.2.1 Gestaltungsempfehlung API Design

Die Geschäftsarchitektur der VE, welche eine digitale Behördenleistung anbieten möchte, steuert zusammen mit möglicherweise bereits im Bundesumfeld wiederverwendbaren APIs die Wahl der Vereinbarungen zum Datenaustausch. Für diese Vereinbarungen bietet die API Architektur Bund einen Ordnungsrahmen, bei dem jeweils die Dimensionen API Protokolle, Schnittstellentypen, MEPs, Datenformate und Message-Typen definiert sind.

Die API Architektur Bund empfiehlt bestimmte, in der Praxis häufig anzutreffende Kombinationsgruppierungen von Vereinbarungen zum Datenaustausch. Diese sind in Tabelle 7 beschrieben und als Good Practice zu verstehen. Eine Beschreibung der Merkmale, welche die Eigenschaften festlegen, findet sich in Kapitel 10 der Hauptversion.

API Protokoll	Schnittstellentyp	MEP	Datenformat	Message-Typ
HTTP	Ressourcenorientiert Methodenorientiert	Request-Response	Keine Einschränkung	Request Response
SOAP/WSDL	Ressourcenorientiert Methodenorientiert	Keine Einschränkung	XML	Request Command Response
gRPC	Ressourcenorientiert Methodenorientiert	Keine Einschränkung	Binärformat[17]	Keine Einschränkung
WebSockets	Ressourcenorientiert	Keine Einschränkung	Keine Einschränkung	Keine Einschränkung
MQTT	Ressourcenorientiert	Messaging	Keine Einschränkung	Request Response Event
AMQP	Ressourcenorientiert	Messaging	Binärformat[dito]	Request Response Event

Tabelle 7: Good Practices zu Vereinbarungen zum Datenaustausch

Im Kontext der Gestaltungsempfehlung API Design empfiehlt die API Architektur Bund RESTful APIs für in digitale Behördenleistungen einzubindende ressourcenorientierte Fachservices. RESTful APIs sind ressourcenorientiert und erfüllen die Bedingungen des Representational-State-Transfer-Architektur-Stils (REST).[18] Dazu gehören alle Kombinationen der Vereinbarungen zum Datenaustausch in Tabelle 7, welche die Bedingungen des REST-Architektur-Stils erfüllen. In Abhängigkeit vom Geschäftsfall können auch andere Architekturstile für APIs verwendet werden, insofern ein akzeptables Kosten-/Nutzen-Verhältnis und Akzeptanz der APIs am Markt gewährleistet sind.

Eine weitere Empfehlung der API Architektur Bund im Kontext API Design ist die Gestaltung des API Gateways in so einer Weise, dass dieser dem API Client bei fehlerhaft formulierten Zugriffen und Nicht-Verfügbarkeit der digitalen Behördenleistung und/oder zugehöriger notwendiger Services in jedem Fall aussagekräftige Fehlermeldungen zukommen lässt.

4.2.2 Gestaltungsempfehlung API Gateway

Die API Architektur Bund empfiehlt grundsätzlich den Einsatz eines API Gateways. Das API Gateway kann verschiedene Ausprägungen haben. Insbesondere kann es als API Proxy oder als API Hub gestaltet sein. Je nach Ausgestaltung kommen verschiedene MEPs zur Anwendung. So wird die Kommunikation bei einer Ausprägung als Proxy über ein synchrones MEP abgewickelt, wohingegen bei einer Ausprägung als API Hub ein asynchrones MEP zur Anwendung kommt und die Messages im API Hub zwischengespeichert werden.

Die API Architektur Bund stellt in Kapitel 9.2 der Hauptversion synchrone Kommunikation und asynchrone Kommunikation mit Hilfe von MEP-Beispielen einander gegenüber. Die Ausprägung des API Gateways hat Auswirkungen auf die MEPs zwischen API Client und API Gateway und zwischen API Gateway und Fachservice. Die MEPs müssen nicht identisch sein. Ist eines der beiden MEPs asynchron, so kommt wegen der Zwischenspeicherung ein

[17] Typisierte Daten, Strings können formatierte Inhalte sein, z.B. XML oder JSON
[18] Die Bedingungen für REST sind in der Hauptversion der API Architektur Bund in Kapitel 8 sowie in der gängigen Literatur zu finden.

API Hub zum Einsatz. Eine Ausprägung eines asynchronen MEPs ist Publish/Subscribe, bei dem vom Server generierte, abonnierte Events bzw. Messages vom API Client empfangen werden. Dies erfordert bidirektionale Kommunikation.

Die API Architektur Bund empfiehlt das MEP Publish/Subscribe nur mit Partner APIs zu kombinieren, da bei diesen Registrierungsanträgen geprüft werden und so die Anzahl abonnierender API Clients gesteuert werden kann. Bei Anwendung von Publish/Subscribe bei Public APIs gibt es keine Prüfung von Registrationsanträgen, so dass der API Anbieter keine Kontrolle über die Anzahl abonnierender API Clients hat. In diesem Fall kann eine MEP-Mischform eingesetzt werden, welche zwischen API Hub und Fachservice Asynchronität zulässt und zwischen API Client und API Hub ein (synchrones) Polling-Verfahren vorsieht.

Weiterhin empfiehlt die API Architektur Bund, Anforderungen, die einen Daten- resp. Informationsfluss vom Fachservice in Richtung des API Clients erfordern, umzusetzen, indem der API Client anstatt des Servers die Kommunikation beginnt.[19]

4.2.3 Gestaltungsempfehlung Identity und Access Management

Identity- und Access Management (IAM) ist eine wesentliche Komponente für integriertes, durchgängiges und elektronisches E-Government, insbesondere, wenn digitale Behördenleistungen über APIs angeboten werden. IAM ist bei Partner APIs und bei Public APIs mit Registrierung notwendig, also in Fällen, wo die Identität mittels Authentifizierung vor dem Zugriff überprüft und die Autorisierung des Zugriffs geprüft wird. Bei Partner APIs ist i.d.R. eine höhere Qualität der Authentisierung erforderlich, welche durch das Level of Assurance (LoA[20]) definiert wird, dessen Einhaltung die Geschäftsfähigkeit IAM bei Zugriffen von Identitäten auf Ressourcen sicherstellen muss. Bei Partner APIs werden Zugriffsanträge immer geprüft und können akzeptiert oder abgelehnt werden, basierend auf den Rollenzuweisungen der Identitäten im IAM.

Die Gestaltungsempfehlung IAM setzt auf der Annahme auf, dass ein API Gateway vorhanden ist und beinhaltet die in den nachfolgenden Abschnitten beschriebenen Elemente.

4.2.3.1 API Zugriff mittels Secure Token und Single-Sign-On

Die API Architektur Bund empfiehlt, Authentisierung und Autorisierung bei API Zugriffen auf Public APIs und Partner APIs mittels Secure Tokens zu implementieren. Mit diesen können Authentisierungs- und Autorisierungsinformationen vom API Client zum API Gateway und vom API Gateway zum Fachservice zu übertragen werden, wodurch das permanente Speichern von Credentials im API Client und Server vermieden werden kann. Durch Verwendung von Secure Tokens kann serverseitiges Session-Management vermieden werden (keine permanenten Sessions), da die Prüfung der Secure Tokens via IAM Service erfolgt. Zudem sind die verbreiteten Standards SAML[21] und OAuth/OIDC[22], welche Secure Tokens definieren, mit HTTP einfach nutzbar. Das Secure Token wird entweder für den API Client allein, oder für den API Client und das API Gateway bereitgestellt, wobei API Client und API Gateway zwei verschiedene Secure Tokens erhalten. Das API Gateway darf das Secure Token des API Clients nicht für den Zugriff auf den Fachservice verwenden. Das Secure Token hat die folgenden Eigenschaften:

- Es stellt eine Ableitung des Credentials einer bekannten Identität dar.
- Es enthält die Bestätigung, dass eine Identität authentifiziert wurde.
- Es kann Autorisierungsinformationen tragen.

Ein Secure Token kann Autorisierungen für mehrere Fachservices enthalten, welche Fachservices im Rahmen derselben digitalen Behördenleistung sind. In der API Architektur Bund ist Single-Sign-On (SSO) dann gegeben, wenn der API Client bei einer digitalen Behördenleistung in einem definierten Zeitfenster für den Zugriff auf mehrere Fachservices nur beim Zugriff auf den ersten Fachservice authentifiziert werden muss.[23] Die API Architektur Bund empfiehlt, in den IAM Services SSO zu implementieren.

4.2.3.2 IAM Bund Rahmenarchitektur

Die API Architektur Bund empfiehlt föderiertes IAM zu verwenden, um bestehende Identitäten und API Rollen im Bundesumfeld für mehrere digitale Behördenleistungen verwenden zu können. Die *IAM Bund Rahmenarchitektur*[24], auf welcher die API Architektur Bund an dieser Stelle aufsetzt, unterscheidet zwischen **föderiertem IAM**, bei

[19] Bei Datenflüssen, wo sich aus wirtschaftlichen Gründen Entwicklung/Betrieb eines APIs nicht lohnen, kann auf alternative Instrumente wie Email zurückgegriffen werden, z.B. bei der jährlichen Erinnerung eines Geschäftspartners an die Erfüllung einer behördlichen Pflicht.
[20] https://www.ech.ch/de/standards/60593
[21] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
[22] https://openid.net/connect
[23] Voraussetzung dafür ist, dass die Identität bei der Authentifizierung die LoA-Anforderungen aller Fachservices erfüllt.
[24] https://intranet.dti.bk.admin.ch/isb_kp/de/home/themen/iam-bund/dokumente-steuerung-iam-bund.html

dem die IAM Services vom API Gateway getrennt und über föderierte IAM-Teil-Services verteilt sind und **nicht-föderiertem IAM**, bei dem das IAM System im API Gateway integriert ist.

Aufgrund des in der IAM Bund Rahmenarchitektur definierten Umgangs mit Secure Tokens bei föderiertem IAM sowie der mit Secure Tokens verbundenen Vorteile ist die Gestaltungsempfehlung IAM auf API Protokolle anwendbar, welche den Umgang mit Secure Tokens beherrschen, was bei Verwendung von HTTP gegeben ist.

4.2.3.3 Arten der Zugriffskontrolle

Die API Architektur Bund empfiehlt bei der Umsetzung der IAM Fähigkeiten einerseits zwischen Nutzung durch das (anonyme) Subjekt und Nutzung durch den (identifizierten) Geschäftspartner und andererseits zwischen Zugriffskontrolle im API Gateway und Zugriffskontrolle im Fachservice zu unterscheiden. Gemäss Tabelle 8 ergeben sich so vier verschiedene Ausprägungen der IAM-relevanten Architektur.

	Client Nutzung durch (anonymes) Subjekt	Client Nutzung durch API Geschäftspartner
Zugriffskontrolle beim API Gateway	Ausprägung 1 Client-Nutzung durch (anonymes) Subjekt mit Zugriffskontrolle beim API Gateway	Ausprägung 2 Client-Nutzung durch API Geschäftspartner mit Zugriffskontrolle beim API Gateway
Zugriffskontrolle beim Fachservice	Ausprägung 3 Client-Nutzung durch (anonymes) Subjekt mit Zugriffskontrolle beim Fachservice	Ausprägung 4 Client-Nutzung durch API Geschäftspartner mit Zugriffskontrolle beim Fachservice

Tabelle 8: Ausprägungen der IAM-relevanten API Architektur

Die beiden Arten der Zugriffskontrolle und die beiden Arten der Client-Nutzung sind zusammen mit detaillierter Beschreibung der Ausprägungen 1 & 4 in Kapitel 8.4 der Hauptversion der API Architektur Bund beschrieben, in Abstimmung mit der IAM Bund Rahmenarchitektur.

4.2.4 Gestaltungsempfehlung API Integration

Geschäftsprozesse benötigen Fachservices, auf welche bei digitalen Behördenleistungen von API Clients via APIs zugegriffen wird, wodurch Fachservice-Funktionen in den API Client integriert werden. Wenn dabei vom Fachservice weitere Fachservices aufgerufen werden, mittels eines mehrstufigen und/oder verzweigten Baumes an API Aufrufen, dann wird dies in der API Architektur Bund als «API Integration im Fachservice des Bundes» bezeichnet.

Die API Architektur Bund empfiehlt, getreu dem „Once-Only-Prinzip“ die Durchgängigkeit von digitalen Behördenleistungen durch API Integration im Fachservice des Bundes sicherzustellen. Geschieht ein solcher Aufruf Bundes-Netzzonen-Grenzen-überschreitend (IP4) oder VE-übergreifend (IP3) wird der Einsatz eines API Gateways empfohlen. Im Allgemeinen sollte die Aufrufkette aus Leistungs-, Wartungs- und Testgründen so kurz und einfach wie möglich gehalten werden.

Bei einer Kette von API Aufrufen mehrerer schreibender Operationen ist Transaktionssicherheit besonders wichtig. Die API Architektur Bund unterscheidet zwischen «Parallele, schreibende Operationen» und «Kaskadierte, schreibende Operationen», was in Kapitel 9.4 der Hauptversion detailliert beschrieben ist. Die API Architektur Bund empfiehlt, um die zur Gewährleistung von Transaktionssicherheit notwendige Komplexität tief zu halten, auf «Parallele, schreibende Operationen» zu vermeiden.

Falls im Rahmen der API Integration geschützte, nicht anonymisierte Daten von anderen VE genutzt werden, muss eine vorgängige datenschutzrechtliche Abklärung erfolgen und eine Einwilligung bei der Datenerhebung eingeholt werden, oder alternativ wird der Verwendungszweck in der jeweiligen gesetzlichen Grundlage ergänzt.

Prinzipiell kann die API Integration auch im API Client des Geschäftspartners stattfinden, falls die digitale Behördenleistung des Bundes nur eine Teil-Leistung der digitalen Leistung des Geschäftspartners ist.

Bei der API Integration orchestriert eine vermittelnde Systemkomponente das Durchlaufen eines mehrstufigen und verzweigten Baums an API Aufrufen. Voraussetzung dafür ist immer, dass die einzelnen Teil-APIs von der vermittelnden Systemkomponente über das Netzwerk angesprochen werden können. Die vermittelnde Systemkomponente kann eine Integration Middleware sein (z.B. WSO2[25]), auf einem Data Service Framework basieren (z.B. ein mehrere Datenquellen integrierendes GraphQL API) oder eine anderweitige Individualentwicklung mit API Unterstützung sein. Diese Aufzählung ist nicht abschliessend. Die Grenzen zwischen dem API Gateway, der API Integration und der API Schicht sind je nach den gewählten Software-Produkten fliessend.

[25] https://wso2.com

4.2.5 Gestaltungsempfehlung API Versioning

Für die Geschäftspartner hat die Versionierungsstrategie von APIs eine zentrale Bedeutung, denn sie schafft Klarheit darüber, ob eine Änderung am API die API Integration unterbricht oder ob über Versionen hinweg ein konsistentes API Verhalten erwartet werden darf. Die API Architektur Bund unterscheidet zwischen der Versionierung von API Release, API Infrastruktur und API Protokoll, wobei nur Empfehlungen für die Versionierung von API Releases gemacht werden.

4.2.5.1 Versionierung von API Releases, Freiheitsgrade & Technische Abbildung

Die API Architektur Bund empfiehlt bei der **Versionierung von API Releases** den Standard Semantic Versioning 2.0.0[26] (SemVer 2.0.0), welcher auf einer dreistelligen Versionsnummer mit MAJOR, MINOR und PATCH basiert und optional eine „Extension" erlaubt. Die Komponenten der Versionsnummer sind detailliert in Kapitel 9.5 der Hauptversion der API Architektur Bund beschrieben, das Pattern wird hier beispielhaft gezeigt:

Versionsnummer von API Releases in der Versionskontrolle:	MAJOR.MINOR.PATCH(-Extension)
Beispiele:	1.4.1, 2.0.0-rc1

Bei Publikation eines neuen API Releases wird der API Code und/oder die API Konfiguration im Repository im Rahmen der Versionskontrolle mit einem Label versehen (Tagging), welches die Versionsnummer gemäss SemVer 2.0.0 abbildet und in der Online-Dokumentation für die Referenzierung des API Releases verwendet wird. Generell ist dem API Anbieter freigestellt, ob die Stellen MINOR, PATCH und Extension in der Online-Dokumentation publiziert und technisch abgebildet werden. Insbesondere ist dies zulässig, wenn die Evolution eines APIs nur Erweiterungen und Defect Fixes hervorbringt, welche keine Anpassungen des API Clients erfordern. Für die Veröffentlichung von Versionsnummern empfiehlt die API Architektur das folgende Pattern:

Veröffentlichte Versionsnummern von API Releases:	(v)MAJOR(.MINOR.PATCH-Extension)
Beispiele:	v2, 1.2, v1.3.3

Die technische Abbildung der Versionsnummer kann beispielsweise in der URL oder in einem HTTP Header sein. Die in der Online-Dokumentation publizierten Versionsnummern manifestieren sich in den publizierten API Releases derart, dass diese für den API Client bei jedem API Request eindeutig erkennbar sind, unabhängig davon wie viele API Releases parallel betrieben werden und wie viele Stellen der Versionsnummer publiziert sind. Ein Überblick mit Beispielen für die technische Abbildung der Versionsnummer ist in Kapitel 9.5 der Hauptversion der API Architektur Bund zu finden.

4.2.5.2 Erzwungene Migration

Soll aus Gründen der Kosteneffizienz eine API Hauptversion ausser Betrieb genommen werden, sollte dem API Geschäftspartner eine am Anwendungsfall orientierte Übergangsfrist für die Umstellung seines API Clients auf eine im Betriebs stehende API Hauptversion gewährt werden.

4.2.5.3 API Gateway und Fachservice

Die API Architektur Bund geht davon aus, dass die Kommunikation zwischen API Client und Fachservice immer über ein API Gateway führt. Das API Gateway besteht dabei aus der API Infrastruktur und einer API Instanz. Der API Release manifestiert sich in der API Instanz des Gateways und der API Instanz des Fachservices mit den beiden Systemkomponenten API Schicht und API Integration.

Die API Architektur Bund empfiehlt, in der Online-Dokumentation immer die Version des API Releases und damit die fachliche Versionierung abzubilden, da es der API Release mit den genannten Systemkomponenten ist, der sich auf die Implementation des API Clients auswirkt. Es ist darauf zu achten, dass ein Anheben der Version der API Infrastruktur auf die API Releases möglichst keine Auswirkungen auf den API Client hat.

[26] https://semver.org