Bundesministerium
des Innern
und für Heimat

bundID

BundID (Nutzerkonto des Bundes)

Basisdienst ID und Postfach-Nachricht

Überblick über das Dokument

Name und Version des Dokuments	BundID (Nutzerkonto des Bundes) Basisdienst ID und Postfach-Nachricht
Aktenzeichen	
Ersetzt	
Zweck des Dokuments	Beschreibung der Schnittstellen zum BundID Identity Provider sowie Beschreibung der Schnittstelle für den Nachrichtenversand an das Postfach in der BundID
Hauptadressaten / Anwendungsbereich	
Weitere Adressaten	
Herausgebende Stelle	Bundesministerium des Innern und für Heimat DVI5
Gültig ab:	20.09.2019
Fortführende Stelle:	
Geplante Fortschreibung:	
Geplante Inhalte der Fortschreibung:	

Zusammenfassung

Im ersten Teil dieses Dokumentes (Kapitel 1-8) werden die derzeit für Fachportale existierenden Schnittstellen zum BundID Identity Provider beschrieben. Über diese Schnittstelle können Online Leistungen und Fachverfahren die Authentifizierung von Nutzer:innen über die BundID ansteuern.

Im zweiten Teil des vorliegenden Dokumentes (Kapitel 9-14) wird die Schnittstelle für den Nachrichtenversand an das Postfach in der BundID beschrieben. Diese Schnittstelle bietet externen Online Leistungen und Fachverfahren die Möglichkeit, Nachrichten in das Postfach eines bestimmten Nutzers abzulegen.

Änderungsprotokoll

BundID - Basisdienst ID und Postfach-Nachricht

(Zusammengeführtes Dokument)

Datum	Version	Änderungsgrund
11.12.2019	0.94	Übernahme, Überarbeitung und Zusammenführung der AKDB-Dokumente
18.12.2019	0.95	Endabnahme
06.01.2020	0.97	Endabnahme
08.01.2020	0.98	Endabnahme
09.01.2020	0.99	Endabnahme
16.01.2020	1.0	Freigabe
11.03.2020	1.1	Zusatz UK Schnittstelle
03.04.2020	1.2	ITZBund Abnahme
15.05.2020	1.3	Korrekturen
18.05.2020	1.4	Korrekturen Verweise
05.06.2020	1.5	Korrekturen
22.06.2020	1.6	Korrekturen Verweise
22.09.2020	1.7	Korrektur 1 Schema Dateien
24.09.2020	1.8	Korrektur 2 Schema Dateien
02.10.2020	1.9	Kapitel Sondersituation Unternehmenskonto [..] entfernt, Korrekturen Kap. 11, 12
10.11.2020	2.0	Erweiterungen in Kap. 5 und Kap. 6
13.11.2020	2.01	Korrekturen Kap 5. und Kap. 6
11.02.2021	2.02	Überarbeitung Kap 7.2 und Kap 8.4
16.02.2021	2.03	Korrekturen

11.03.2021	2.04	Entfernung „Künstlername“; Einarbeitung STORK-QAA-Level-3; Überarbeitung BSP-Quittung
09.04.2021	3.01	Ergänzung um Changes aus Release 3.1.0, redaktionelle Korrekturen, Entfernung Organisationskonto, Aufnahme Kap. 5.2 und 5.3, Anpassung Attribute in Kap. 6.1 und entspr. Hinweise, Entfernung Kap. 7, Kap. 8.3 ist obsolet, Aufnahme OZG-Hinweis in Kap. 11.2.5, Aufnahme Kap. 13 und 14
20.04.2021	3.02	Qualitätssicherung
30.04.2021	3.03	Korrektur technischer Begriff Postfach zu Postkorb
20.05.2021	3.2.0	Ergänzung um Changes Release 3.2.0.0, Kapitel 11.1.2, Kapitel 11.2.6 (Tabelle 9005), Kapitel 11.2.7
06.08.2021	3.2.1	Ergänzung um Changes Release 3.2.1.0, Kapitel 6.1, Kapitel 6.2.2, Kapitel 7.2, Kapitel 11.2.4
09.09.2021	3.2.1.1	Hotfix
16.09.2021	3.2.1.2	Hotfix und Aktualisierung 15.6.1 bspnachrichten-2.13.xsd
16.11.2021	4.0.0.0	Anpassung zu Release 4.0.0.0, Kapitel 6.2.6; Kapitel 8.3; Kapitel 9 eingefügt
10.12.2021	4.0.1.0	Kapitel 6 Hinweise zu Friendly Names, Kapitel 9.2 eingefügt
21.12.2021	4.0.1.0	Anpassung Logo und Bezeichnung BMI, Qualitätssicherung
04.04.2022	5.0.0.0	Überarbeitung Kapitel 8.3 und Kapitel 9 gesamt
05.05.2022	N/A	Qualitätssicherung
02.06.2022	5.0.2.0	Überarbeitung 6.1 und 9.1, Einarbeitung Kapitel 12.1.3
07.07.2022	N/A	Qualitätssicherung

12.08.2022	N/A	Ergänzung um Kapitel 10 SAML Beispiele, Überarbeitung Kapitel 11.2, Trennung SAML-Response Beispiele als Anhang 1 zur Schnittstellenbeschreibung
18.08.2022	N/A	Qualitätssicherung
27.01.2023	N/A	Redaktionelle Änderungen, Kapitel 9.4 eingefügt, Korrekturen Kap. 5.4 und 6.2.5
23.02.2023	N/A	Entfernung Kapitel 17.4, Qualitätssicherung
04.05.2023	N/A	Link unter Kapitel 6.1 Seite 20 aktualisiert
07.08.2023	N/A	Entfernung bPK, Version 1, Überarbeitung Kapitel 5.1, 6.1, 6.2.2, 7.2, 9.2, 9.4, 9.5
17.10.2023	N/A	Layout
16.11.2023	N/A	Überarbeitung Kapitel 9.2.1, 9.2.2, 9.4, Erweiterungen in Kapitel 6.1, 6.2.1, 9., 9.5, Aufnahme Kapitel 9.6 und 9.7

Inhaltsverzeichnis

1 Einleitung

Die BundID bildet über eine offene Infrastruktur mit den beteiligten Systemen den Portalverbund.

Die Bürger:innen legen sich eine individuelle BundID zu und verwalten dort Ihre persönlichen Daten und Zusatzinformationen. Die Befüllung der personenbezogenen Datenfelder für die BundID erfolgt entweder über die freiwillige Angabe (bei Nutzung von Benutzername/Passwort-Paar) oder automatisiert bei der erstmaligen Nutzung z.B. mittels des Online-Ausweises.

Fachportale als Drittanwendungen im Portalverbund binden sich über definierte Standardprotokolle mittelbar an das Bundesportal an und profitieren im Sinne der Datensparsamkeit von der zentralisierten Verwaltung der personenbezogenen Nutzerdaten in der BundID.

In diesem Dokument werden die derzeit für Fachportale existierenden Schnittstellen zum BundID Identity-Provider beschrieben.

2 Das Bundesportal – eine offene Infrastruktur für Fachportale öffentlicher Verwaltungen

Ausgewählte personenbezogene Daten aus dem Basisdienst „Nutzerkonto“ stehen im Rahmen einer offenen Infrastruktur auch einem weiteren Kreis an Drittanwendungen außerhalb des eigentlichen Bundesportals zur Verfügung (s. Kp. 8 - Entscheidungsunterstützung und Handreichungen).

Diese offene Infrastruktur basiert auf international anerkannten, quell-offenen Standardtechnologien [1] und sichert die Drittanwendungen gegen unberechtigte Zugriffsversuche (d. h. ohne vorherige Autorisierung durch die ordnungsgemäßen Benutzer) ab. Die von den Bürgerinnen und Bürgern dafür zu nutzenden Authentisierungsinformationen (u. a. Benutzername/Passwort, ePA-Pseudonym) sind identisch mit jenen, die für die BundID im Bundesportal hinterlegt sind, so dass eine erneute Registrierung in den angebundenen Drittanwendungen hinfällig ist.

Die an der BundID teilnehmenden Online Leistungen werden dann „Service-Provider“ genannt und verbleiben daher nicht nur in der rechtlichen, sondern auch in der operativen Zuständigkeit derjenigen Körperschaft, in deren Namen die Dienstleistung den Bürgerinnen und Bürgern angeboten wird. Der Fortbestand eigener Web-Angebote erweist sich insbesondere in jenen Fällen als sinnvoll, in denen ein hohes Maß an systemischer Komplexität gefordert oder das vorrangige Augenmerk auf den Schutz von langfristig getätigten, strategischen IT-Investitionen gelegt wird.

Die Auslagerung des Zugriffsschutzes der Drittanwendungen an Komponenten des Portalverbunds erfordert allerdings einen, je nach Technologiestand des Service-Providers, nicht geringen Eingriff in die bisherige Funktionsweise der Web-Anwendung. Aus technologischer Sicht lassen sich neu aufzubauende oder bestehende Applikationen die bereits mit ähnlichen Standards wie OASIS SAML versehen sind, mit geringerem Aufwand in den Portalverbund aufnehmen als jene Anwendungen, die anders geartete Zugriffskontrollmechanismen (wie z. B. Kerberos) nutzen.

3 Übersicht über die Funktionsweise der offenen Infrastruktur

Im Detail stellt die Übersichtsgrafik den Authentisierungs- und Autorisierungsvorgang von Dienstnutzern im Zusammenspiel mit der offenen Infrastruktur und einer daran angebundenen Drittanwendung dar. Der abgebildete Ablauf in sieben Teilschritten folgt den international weitverbreiteten Standards der OASIS Group, wie sie auch im Kern der eCard-API des bundesdeutschen elektronischen Identifikationsmittel (ePA, eAT, EU-Karte) zum Einsatz gelangen (https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03130/TR-03130_node.html).

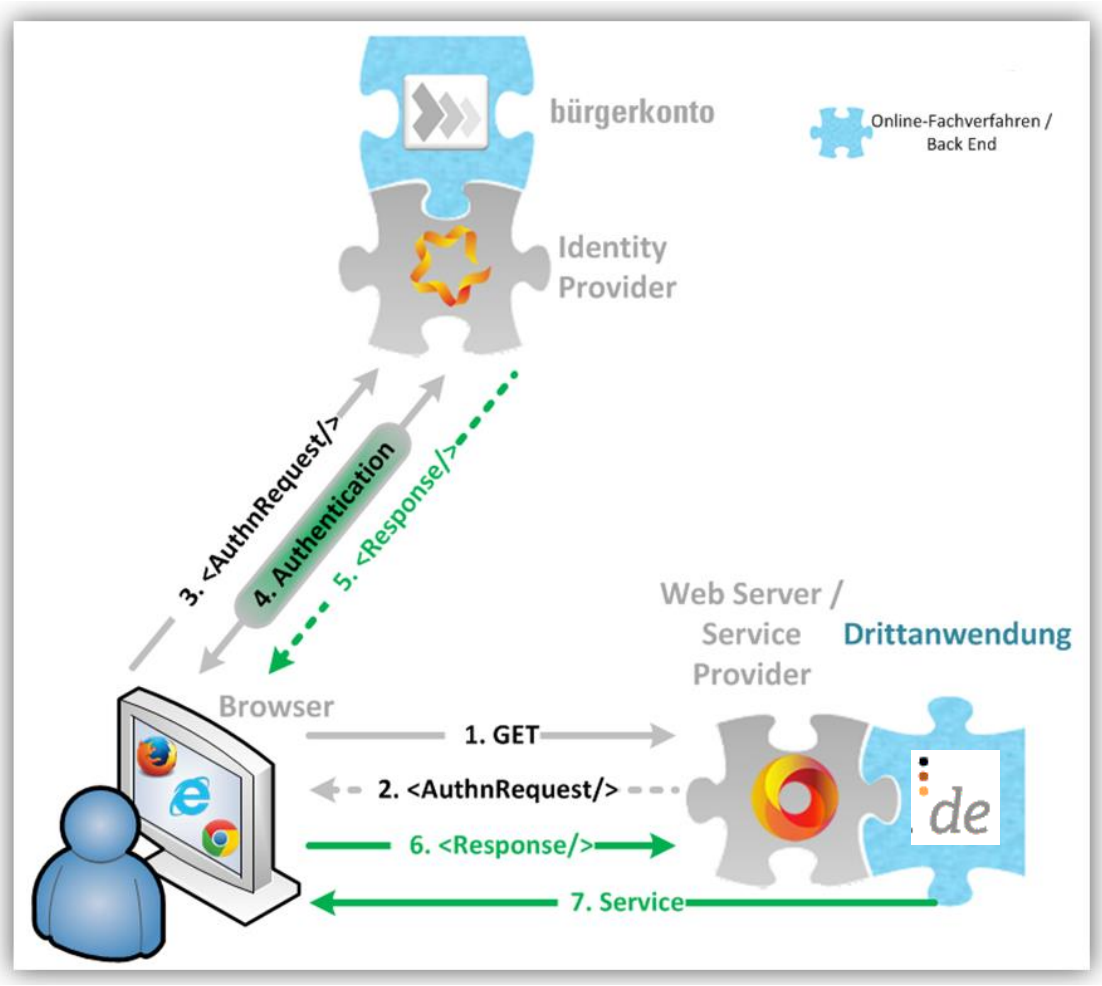


Abbildung 1: Zusammenspiel der Komponenten in der offenen Infrastruktur des Portalverbunds

Die einzelnen Teilschritte sind nur vereinzelt für die Dienstnutzer wahrnehmbar (wie aus den Screenshots ersichtlich) und erleichtern so die Handhabung des komplexen Authentisierungs- und Autorisierungsvorgangs:

Solange der Benutzer nicht authentisiert wurde, kann lediglich auf die öffentlich zugänglichen Bereiche der Service-Provider-Applikation zugegriffen werden. Sobald eine zugriffsgeschützte Ressource im Web-Angebot der Drittanwendung angefragt werden sollte, wird der Zugriffsversuch auf Seiten des Service-Providers abgefangen (Schritt 1).

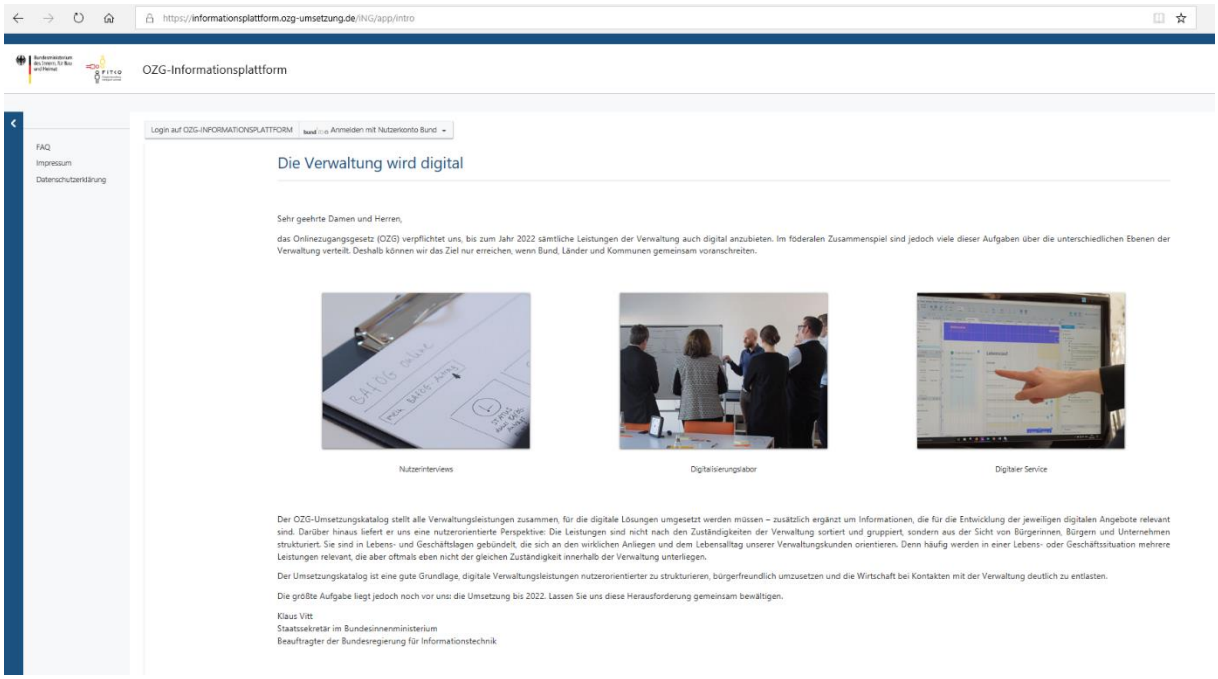


Abbildung 2: Öffentlich zugänglicher Bereich der Service-Provider-Applikation

Seitens des Fachportals wird daraufhin eine Authentisierungsanfrage an den BundID Identity-Provider als Authentisierungsinstanz gesendet und der Benutzer dorthin umgeleitet. Derzeit geschieht dies in Form eines OASIS SAMLv2-konformen Authentication Requests via HTTP-REDIRECT oder HTTP-POST über den Browser des Benutzers (Schritt 2). Diese Anfrage wird nicht direkt an den Identity Provider versendet, sondern mittelbar über den Browser des Dienstanfragers geleitet (Schritt 3).

Der Identity-Provider fordert nun den Dienstanfrager auf, sich ihm gegenüber durch ein vorkonfiguriertes Authentisierungsverfahren als bekannter Benutzer zu erkennen zu geben: Dies kann z. B. durch den Abgleich von Benutzername und Passwort mit den im Bürgerkonto hinterlegten Daten oder durch die Nutzung der Online-Ausweisfunktion erfolgen. Erst im Zuge einer positiv durchlaufenen Authentisierung (Schritt 4) erfolgt am Identity Provider die Erstellung einer Antwortnachricht, die unter Hinzuziehung der in der BundID für die Person bzw. die Organisation hinterlegten Daten erweitert wird. Ab diesem Zeitpunkt kann der Dienstnutzer im rechtlichen Sinne als authentisierter Benutzer im Rahmen der Infrastruktur des Portalverbunds gelten.

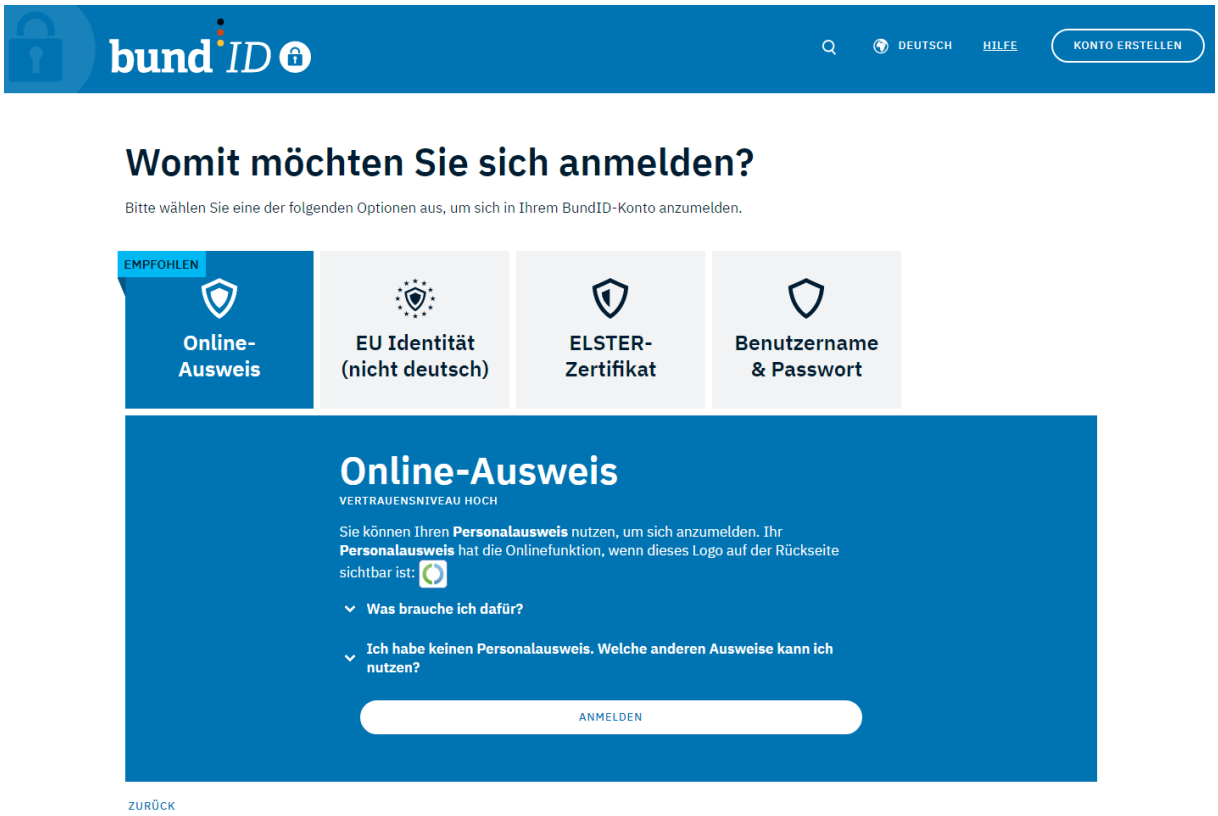


Abbildung 3: Authentisierung

Die erzeugte Antwortnachricht wird im Hintergrund als SAML-Response verschlüsselt und unter Einberechnung von Zertifikatsinformationen signiert (Schritt 5) an den Browser des potentiellen Dienstnutzers zurückgesendet. Von dort aus wird die SAML-Response an die Drittanwendung weiter durchgestellt. Die Verwendung BSI-konformer Verschlüsselungstechnologien [2] im vorherigen Schritt stellt sicher, dass der Dienstnutzer an dieser Stelle keine Möglichkeit zur Vortäuschung oder beabsichtigten Veränderung des Inhalts hat.

Der Service-Provider empfängt die SAML-Response (Schritt 6) und überprüft zunächst die Nachricht auf ihre Authentizität und Integrität hin. Erst dann wird überprüft, ob der Inhalt der vom Identity Provider stammenden Antwortnachricht den regelbasierten Erfordernissen der Drittanwendung genügt. Erst jetzt ist der authentisierte Dienstnutzer im rechtlichen Sinne auch autorisiert und zugriffsbefugt.

Der Service-Provider gewährt daraufhin den Zugriff auf die zunächst geschützte Ressource und führt die ursprüngliche Zugriffsanfrage aus. Die Antwort wird auf dem zugehörigen Anwendungsserver generiert und an den Browser des Dienstnutzers gesendet (Schritt 7).

Für den Zeitraum der weiteren Sitzung wird fortan die Auslieferung aller weiteren zugriffsgeschützte Ressource(n) je nach Anfrage (und applikationsspezifischem Verhalten) erlaubt.

Darüberhinausgehende Zugriffsrestriktionen (z. B. nach RBAC) sind ausschließlich Teil der Applikationslogik des Fachportals und liegen jenseits des Funktionsumfanges der zur Verfügung gestellten Infrastruktur.

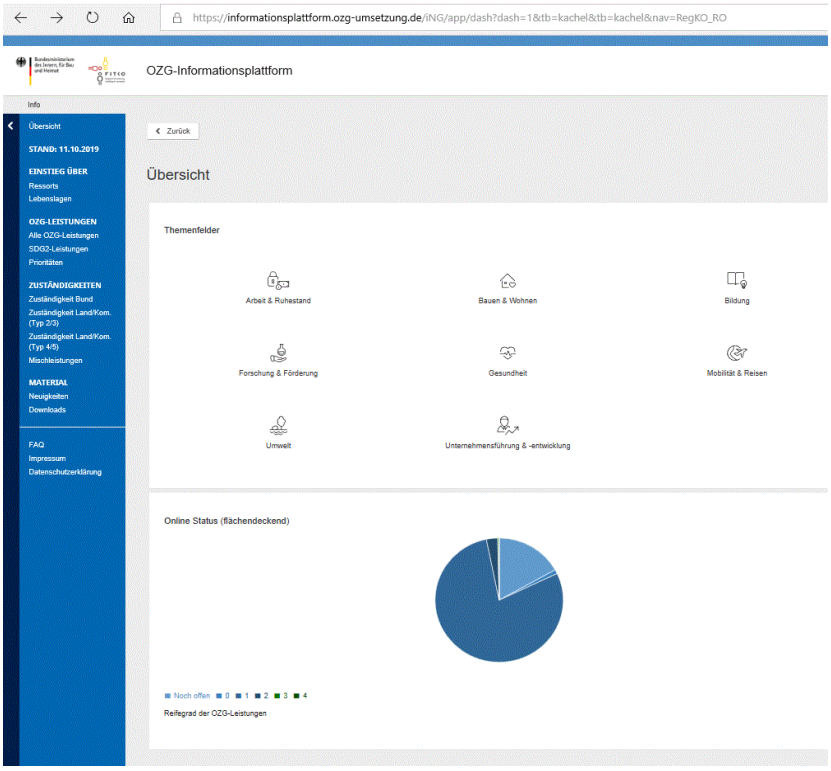


Abbildung 4: geschützte Ressource - OZG-Informationsplattform

4 Unterstützte Anwendungsfälle

Die Online Leistungen der Bundesverwaltung schließen sich mittelbar an die BundID an und lagern Ihre Nutzerauthentisierung an die Identity-Infrastruktur aus. Diese Service-Provider müssen als Endpunkt für OASIS SAML v2-Token i. S. d. Web Browser Single-Sign-On-Profile [3] mit HTTP-REDIRECT oder HTTP-POST-Binding fungieren können.

Dies ist allgemein über zwei Wege möglich:

1. Über direkte Token-Generierung und -Nutzung innerhalb der Drittapplikation (z. B. über Open-Source-Bibliotheken)
2. Über indirekte Token-Generierung und -Nutzung mittels Reverse-Proxy (z. B. über konfigurierbare Implementierungen diverser Hersteller)

4.1 Anwendungsfall 1 – direkte Tokennutzung

Die Drittanwendung kann direkt als Endpunkt für die Erzeugung von Authentication Requests und der Nutzung von SAML-Assertions fungieren. Dafür muss die Anwendung das bereits erwähnte OASIS SAMLv2 Web Browser Single-Sign-On-Profile implementieren.

Bei Wahl dieser ersten Betriebsvariante empfiehlt sich die Nutzung der Open-Source-Bibliothek OpenSAML oder ähnlicher Frameworks. Implementierungsbeispiele zur Nutzung finden sich z.B. im Entwickler-Paket von Governikus-Autent oder in frei verfügbaren Quellen Dritter [4].

4.2 Anwendungsfall 2 – indirekte Tokennutzung

Erlaubt die Rechenzentrumsinfrastruktur den Betrieb von zusätzlichen Komponenten auf dem der Drittanwendung vorgelagerten Web-Server, bietet sich darüber hinaus die Nutzung von sogenannten Reverse-Proxy-Installationen an, die ihrerseits als Endpunkt des SAML-Protokolls agieren und die entsprechenden Requests und Responses für das Fachportal verarbeiten. Der Service-Provider besteht dann aus der eigentlichen Drittanwendung (dem webbasierten Fachverfahren) und dem vorgeschalteten Reverse Proxy als SAML-v2-Endpunkt.

Bei der Wahl dieser zweiten Betriebsvariante bietet sich die Nutzung von Reverse-Proxy-Produkten an, wie sie von Herstellern wie "ForgeRock" [5] oder "Shibboleth Consortium" [6] zur Verfügung stehen. Implementierungs- und Konfigurationsbeispiele zur Nutzung finden sich in den öffentlich zugänglichen Produktdokumentationen der jeweiligen Hersteller und sind einzelfallabhängig je nach Drittanwendung und Rechenzentrumsinfrastruktur zu interpretieren.

Die im SAML-Token enthaltenen Attribute werden in Abstimmung mit den Applikationsarchitekten nach erfolgter Anmeldung eines Benutzers als zusätzliche HTTP-Header-Attribute in die Online-Sitzung injiziert und stehen dann der Drittanwendung im folgenden Verlauf der Sitzung zur weiteren Verarbeitung zur Verfügung.

5 Unterstützte Authentisierungsverfahren und Anfragearten

5.1.1 Die BundID verfügt über folgende Authentisierungsmethoden, die auch ein zugehöriges Vertrauensniveau (s. Kapitel 6.2.1 bPK2 - bereichsspezifisches Personenkennzeichen

Da für die Service Provider das Entpacken und das Verarbeiten der bPK eine unnötige Hürde war, wurde die bPK2 eingeführt.

Das bPK (bereichsspezifisches Personenkennzeichen) dient in Anlehnung an die österreichische Bürgerkarten-Infrastruktur zur datenschutzfreundlichen, für die jeweilige Drittanwendung eindeutig geltenden Identifizierung eines Nutzers.

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)
bPK2	bPK2	urn:oid:1.3.6.1.4.1.25484.494450.3

Bei FINK wird der Identifier aus dem anderen Servicekonto übernommen und mit Prefixen versehen, damit keine Überschneidung mit existierenden bPK2s möglich ist.

Vertrauensniveau

Bezeichnung	**Ausgestaltung**
Benutzername/Passwort (entspricht Nicht-Vertrauensniveau Basisregistrierung, STORK-QAA-Level-1)	• Ein softwarebasiertes Verfahren • Die Registrierung erfolgt einzig durch dieBundID • Attribute sind nicht überprüft
ELSTER Zertifikat (entspricht Vertrauensniveau substanziell, STORK-QAA-Level-3)	• Ein software-basiertes PKI-Verfahren • Die Registrierung und Nutzung erfolgt in Koordination mit den Systemkomponenten von ELSTER. • Attribute sind Melderegister-geprüft
neuer elektronischer Personalausweis (entspricht Vertrauensniveau hoch, STORK-QAA-Level-4)	• Ein hardwarebasiertes PKI-Verfahren • Die Registrierung und Nutzung erfolgt in Koordination mit BSI-TR-zertifizierten eID-Servern • Attribute sind Melderegister geprüft

5.2 Auswahl bestimmter Authentisierungsverfahren

Authentisierungsverfahren können nach Vertrauensniveau gruppiert und im Rahmen des SAML-AuthenticationRequests von einer Drittanwendung angefordert werden (sog. RequestedAuthn-Context). Es wird der Comparison-Qualifier „minimum“ unterstützt. Die implizite Einschränkung von Authentisierungsmethoden über den Qualifier „exact“ ist obsolet und wird explizit durch eine Extension im SAML-Request gelöst.

Perspektivisch ist es geplant weitere Authentifizierungsmittel, die gemäß eIDAS-Verordnung notifiziert wurden, anzubinden. Diese können dann für die Authentifizierung auf den jeweiligen Vertrauensniveau-Stufen verwendet werden.

Für den Fall der indirekten Tokennutzung (s. Kapitel 4.2 Anwendungsfall 2 – indirekte Tokennutzung) müssten die Erweiterungen über produktspezifische Konfigurationen vorgenommen werden, so z.B. beim Reverse-Proxy des Herstellers „Shibboleth Consortium“ mittels Templates zum SessionInitiator [7].

Bei der Authentifizierung mit eIDAS liegt uns momentan noch kein Feedback von Service-Providern vor. Des Weiteren ist noch in der Diskussion, welche Art von ID bei einer temporären Anmeldung (ohne permanentes Nutzerkonto) übergeben wird und wie bzw. ob diese später wieder zugeordnet werden kann.

5.3 Besonderheiten bei der Nutzung von interoperablen Nutzerkonten

Im Rahmen des Föderierten Identitätsmanagements Interoperable Nutzerkonten in Deutschland (FINK), ist es möglich, dass die Authentifizierung durch ein Nutzerkonto eines anderen Teilnehmers von FINK durchgeführt wird. Hierbei wird der SAMLAuthenticationRequests an das ausgewählte Nutzerkonto durchgereicht (siehe FINK Informationsplattform). Das ggfs. vorgegebene Vertrauensniveau wird dabei mitgegeben, sodass das empfangene Nutzerkonto entsprechend reagieren kann. Im Umkehrschluss reagiert das Nutzerkonto der AKDB gleichermaßen.

Die Attributmenge kann je Nutzerkonto eines Teilnehmers variieren. Sofern ein PersonIdentifier mitgeliefert wird, werden diese Attribute aus Datenschutz-Gründen gefiltert und nicht verarbeitet.

Die Interoperabilität von Postfächern wird zu einem späteren Zeitpunkt bereitgestellt, insofern werden potenziell mitgelieferte Postfachreferenzen ebenfalls gefiltert und nicht verarbeitet.

Für weitergehende Informationen wird auf die Informationsplattform des FINK Verbund verwiesen: https://informationsplattform.efink.services/

5.4 Besonderheiten bei der Nutzung der Methode "Temporär Login"

Gemäß OZG muss den Nutzer:innen die Möglichkeit gegeben werden, das Nutzerkonto ohne Langzeitspeicherung von Daten verwenden und sich so gegenüber Drittanwendungen authentifizieren zu können. Hierfür bietet das Nutzerkonto die Methode "als Gast anmelden", welche am IDP als zusätzliche Methode angeboten wird. Für die Gastanmeldung können alle elektronischen Identifikationsmittel verwendet werden, wie bspw. der Online-Ausweis. Die Zugangsmittel Diia und Benutzername&Passwort können hier nicht verwendet werden. Zu beachten ist, dass bei dieser Methode nur Daten ausgelesen und weitergegeben werden. Es erfolgt keine weitergehende Speicherung oder Verarbeitung der Daten. Das hat zur Folge, dass die technischen Nutzdaten nur bedingt bereitgestellt werden. Insbesondere das Postkorb-Handle und das bPK2 entfällt bei dieser Methode.

Sofern eine Drittanwendung ohne diese Daten nicht nutzbar ist, muss das Error-Handling bei der Drittanwendung stattfinden und nicht im Nutzerkonto.

6 Attribute im SAML-Token

Derzeit stellt die Identity-Infrastruktur nachfolgend aufgelistete Attribute aus dem Bürgerkonto zur weiteren Nutzung in Fachportalen im Rahmen von SAML-Assertions zur Verfügung. Die darin übermittelten Daten entsprechen immer dem zum Zeitpunkt der Token-Ausstellung aktuellen Datensatz in der BundID. Die Daten werden im UTF-8-Zeichensatz NFC-kodiert.

Hinweis:

Entsprechend des verwendeten Protokolls (Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0) https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf darf für das Mapping der Attribute nicht das XML Attribut "FriendlyName" verwendet werden, da diese Werte nicht stabil und zudem optional sind. Hierfür ist der SAML2 Formal Name (URN-notiert) zu verwenden, da es sich hierbei um eine gleichbleibende ID handelt.

6.1 Personenbezogene Stammdaten

Die unmittelbar personenbezogenen Stammdaten aus der BundID werden in LDAP-konformer Notation zur Verfügung gestellt. Die Konversion in LDAP-Notation orientiert sich an den IETF-Standards aus RFC 4519 [8] und RFC 4524 [9].

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)	Hinweise
Vorname(n)	givenName	urn:oid:2.5.4.42	
Nachname	surname	urn:oid:2.5.4.4	
Emailadresse (optional)	mail	urn:oid:0.9.2342.19200300.100.1.3	
Strasse	postalAddress	urn:oid:2.5.4.16	
PLZ	postalCode	urn:oid:2.5.4.17	
Wohnort	localityName	urn:oid:2.5.4.7	
Land (Adresse)	Country	urn:oid:1.2.40.0.10.2.1.1.225599	Als ISO 3166-1 alpha-2 gepflegt. Ein D von der eID oder eIDAS wird durch DE ersetzt.

Akad. Titel (optional)	personal-Title	urn:oid: 0.9.2342.19200300.100.1.40	
Anrede	gender	urn:oid:1.3.6.1.4.1.33592.1.3.5	Die Anrede wird als numerischer Wert nach ISO 5218:2004 codiert [11]. "Keine Angabe" = 0 (fälschlicherweise Wert 9 bis NK Release 7) "Herr" = 1 "Frau" = 2
Geburtsda-tum	birthdate	urn:oid:1.2.40.0.10.2.1.1.55	Nach ISO 8601 im sog. extended Format in der Form JJJJ-MM-TT ohne weitere Zeitangabe
Geburtsort (optional)	placeOfBirth	urn:oid:1.3.6.1.5.5.7.9.2	
~~Ausstellen-der Staat~~	~~issuingState~~	~~urn:oid:1.2.40.0.10.2.1.1.552244~~	seit 02/2019 nicht mehr unterstützt
Geburts-name	birthName	urn:oid:1.2.40.0.10.2.1.1.225566	

Staatsangehörigkeit	nationality	urn:oid:1.2.40.0.10.2.1.1.225577	
De-Mail	DeMail	urn:oid:1.3.6.1.4.1.55605.70737875.1.1.1.7.1	
Telefonnummer	telephoneNumber	urn:oid:2.5.4.20	Darstellung ab 05/2021 (R3.2.0.0) als international gültige Telefonnummer; in der BundID ab 09/2021 produktiv
eIDAS-Issuing-Country	eIDAS-Issuing-Country	urn:oid:1.3.6.1.4.1.25484.494450.10.1	Basiert auf SendingMemberState (TR-03130 eID-Server), optional, ISO 3166-1 alpha-2
Dokumententyp	documentType	urn:oid:1.2.40.0.10.2.1.1.552255	Ab Release 8.1, mit folgenden Einschränkungen zur Verfügbarkeit des Dokumententyps: • Wenn es durch ein interoperables SK geliefert wird • Bei einer Gastanmeldung mit eID • Wenn der Wert zukünftig am Nutzerkonto persistiert wird In anderen Fällen steht der Wert nicht zur Verfügung. Mögliche Werte werden in der Technische Richtlinie TR-

			03127 definiert (Verlinkung Stand Oktober 2023).

Bei der Authentifizierung mit eIDAS-notifizierten Identifikationsmittel gilt der Mindestdatensatz nach eIDAS-VO, daher kann es zu einem reduzierten Datensatz kommen. Siehe hierzu:

https://ec.europa.eu/digital-building-blocks/wikis/download/attachments/467109280/eidas_saml_attribute_profile_v1.0_2.pdf?version=1&modificationDate=1639417533738&api=v2

6.2 Technische Nutzdaten

Darüber hinaus werden technische Nutzdaten zur erleichterten Datenpersistenz in Drittanwendungen, zum Kontext des Authentisierungsvorgangs und zu weiteren Schnittstellen (Postfach) bereitgestellt.

6.2.1 bPK2 - bereichsspezifisches Personenkennzeichen

Da für die Service Provider das Entpacken und das Verarbeiten der bPK eine unnötige Hürde war, wurde die bPK2 eingeführt.

Das bPK (bereichsspezifisches Personenkennzeichen) dient in Anlehnung an die österreichische Bürgerkarten-Infrastruktur zur datenschutzfreundlichen, für die jeweilige Drittanwendung eindeutig geltenden Identifizierung eines Nutzers.

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)
bPK2	bPK2	urn:oid:1.3.6.1.4.1.25484.494450.3

Bei FINK wird der Identifier aus dem anderen Servicekonto übernommen und mit Prefixen versehen, damit keine Überschneidung mit existierenden bPK2s möglich ist.

6.2.2 Vertrauensniveau

Informationen über die vom Benutzer gewählte Authentisierungsmethode zur Initialisierung der Sitzung werden mittelbar in Form einer akkumulierten Trustlevel-Angabe (d. h. zum Authentisierungsvorgang und zur Herkunft der Attribute aus der BundID) verfügbar gemacht.

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)
Vertrauensniveau	EID-CITIZEN-QAA-LEVEL	urn:oid: 1.2.40.0.10.2.1.1.261.94

Aufgrund der strategischen Wichtigkeit des von der Europäischen Kommission geförderten Large-Scale-Pilot-Projekts STORK [11] und im Lichte der eIDAS-Verordnung [12], wurde bis zur Veröffentlichung entsprechender Technischer Richtlinien seitens des BSI der Trustlevel-Ansatz nach STORK-Methodik gewählt [13]. Zum derzeitigen Zeitpunkt unterscheidet die Identity Infrastruktur folgende Trustlevel nach STORK, die zukünftig als Kategorien mehrere gleichrangige Authentisierungsmethoden beinhalten können:

Bezeichnung	Bedeutung für Drittanwendungen
STORK-QAA-Level-1	aktuelle Authentisierung mittels Benutzername/Passwort; registrierte Attributdaten ohne hoheitliche Prüfung (= selbstregistrierte BundID bzw. eIDAS-Äquivalent)
STORK-QAA-Level-3	Aktuelle Authentisierung mittels ELSTER-Zertifikat; Registrierte Attribute aus dem ELSTER Zertifikat (= ELSTER Zertifikat registrierte BundID bzw. eIDAS-Äquivalent)
STORK-QAA-Level-4	aktuelle Authentisierung mittels Online-Ausweisfunktion (ePA, eAT, EU-Karte); registrierte Attributdaten aus Ausweismittel (= eID-registrierte BundID bzw. eIDAS-Äquivalent)

6.2.3 Version

Das Attribut Version dient als Vorbereitung um Änderungen besser kommunizieren zu können. Damit können Service-Provider technisch feststellen, mit welcher fachlichen Version der Schnittstelle sie es zu tun haben und können besser darauf reagieren. Die initiale Version lautet `2020.2.1` und als Konvention wird Calendar Versioning [14] verwendet.

6.2.4 AssertionProvedBy

Darüber wird die Quelle transportiert, die die Identität überprüft hat. Im ersten Schritt wird hier eIDAS bei der Authentifizierung über eIDAS übermittelt. In den nächsten Versionen werden die weiteren IDs auch in der Schnittstelle übergeben.

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)
AssertionProvedBy	AssertionProvedBy	urn:oid:1.3.6.1.4.1.25484.494450.2

Liste der möglichen IDs:

- eIDAS
- eID
- Smart-eID
- Elster
- Benutzername
- FINK
- Diia

6.2.5 Postkorb-Handle

Das Postkorb-Handle in seiner jetzigen Form kann als Eingabeparameter für die Zustellung von Postfachnachrichten genutzt werden. Um Nachrichten einem bestimmten Vertrauensniveau in der späteren Ansicht zuzuordnen, ist bei der Benutzung der eigenen Postfach-API zusätzlich noch das Vertrauensniveau mitzugeben.

Dieses Attribut ist nicht dafür geeignet, als eindeutige Nutzerreferenzierung in der Drittanwendung verwendet zu werden.

Bezeichnung	FriendlyName	SAML2 Formal Name (URN-notiert)
Postkorb-Handle	legacyPostkorbHandle	urn:oid:2.5.4.18

Jeder BundID ist immer ein Postkorb-Handle zugeordnet, welches wiederum genutzt werden kann, um ein Postfach eindeutig zu referenzieren.

Weiterführende Informationen zur Nutzung dieses Attributs bei der Kommunikation mit einem Postfach können den Kapiteln 9 ff. entnommen werden.

7 Betriebsvoraussetzungen

7.1 Infrastruktur

Das ITZBund als Betreiber der SAML-basierten Identity-Infrastruktur der BundID sichert die dauernde Erreichbarkeit folgender Komponenten zu:

- Identity-Provider für die unterstützten Authentisierungsmethoden

Voraussetzung zur Teilnahme an der SAML-basierten Identity Infrastruktur ist:

- Einreichung der Kooperationsvereinbarung,
- Bereitstellung der SAML-Metadaten seitens der Drittanwendung.

Es ist darauf zu achten, dass

1. die einzureichenden Metadaten *kein Ablaufdatum* enthalten (validUntil-Attribut),
2. die Metadaten *kein ID-Attribut* enthalten,
3. die *entityID als URI in URL-Notation* mit https-Protokoll-Prefix ohne Portnummer anzugeben ist. Diese muss nicht(!) zwingend mit der tatsächlich genutzten URL-Domain übereinstimmen, ist aber in Zusammenhang mit dem Attribut bPK2 in dieser Notation anzugeben. Die Auswahl der entityID kann nach Aufnahme in den Wirkbetrieb nicht(!) mehr verändert werden und sollte daher den Betreiber der Drittanwendung eindeutig identifizieren (also nicht unspezifisch sein) wie folgendes Negativbeispiel „https://drittanwendung.com/serviceprovider“.

Erst nach erfolgter Bereitstellung der SAML-Metadaten kann dem Teilnehmer die Metadaten des Identity Providers (bzw. für Drittanwendungen innerhalb des ITZBund-RZ die URL zu den Federation-Metadaten) mitgeteilt werden.

Die Bekanntgabe der Metadaten des Identity Providers durch das ITZBund ist als technologische Schnittstelle hinreichend für die Entwicklung einer Drittanwendung.

Für die Integrationsumgebung wird auch die IP-Adresse des zu verbindenden Verfahrens benötigt. Für die Produktionsumgebung ist dies nicht nötig.

7.2 Zusätzliche Metadaten

Diese Informationen können in den Metadaten übergeben und dem Nutzer angezeigt werden, bevor die Daten über die SAML-Response transportiert werden.

Falls die Informationen nicht zur Verfügung stehen, werden dem Nutzer keine Informationen über den Dienst angezeigt, an den er nach seiner Zustimmung die Daten übermittelt. Die Informationen sind also technisch nicht relevant, bieten aber einen fachlichen Mehrwert.

Auszug aus den Metadaten:

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="https://example.com/sp">
    <md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:Extensions>
    <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
    <mdui:DisplayName xml:lang="de">Beispielanwendung</mdui:DisplayName>
    <mdui:Description xml:lang="de">Die Beispieldomain example.com ist eine Second-level-Domain, die von der
    Internet Engineering Task Force permanent reserviert wurde.</mdui:Description>
    <mdui:InformationURL xml:lang="de">https://example.com/wir-ueber-uns</mdui:InformationURL>
    <mdui:PrivacyStatementURL xml:lang="de">https://example.com/datenschutz</mdui:PrivacyStatementURL>
    </mdui:UIInfo>
```

Nach erfolgreicher Entwicklung und Test der Anbindung Ihrer Verwaltungsleistung an der BundID, erklären Sie die Fertigstellung auf Ihrer Stage-Umgebung und stellen dem BMI einen Zugang zur Verfügung. Nach abschließender Sichtung durch das BMI werden die Metadaten Ihrer Produktivumgebung in der Produktivumgebung der BundID freigeschaltet, so dass Sie Ihre Produktivumgebung entsprechend konfigurieren und testen können. Sie können ab diesem Zeitpunkt selbst entscheiden, wann Sie die Verwaltungsleistung produktiv setzen. Beachten Sie dabei, dass eine Produktivsetzung das Vorhandensein von Datenschutzkonzept, IT-Sicherheitskonzept und Vereinbarung zur Auftragsdatenverarbeitungvoraussetzt.

8 Entscheidungsunterstützung und Handreichungen

8.1 Architekturüberlegungen bei Auswahl eines SAML-Bindings

Für die Anbindung einer Drittanwendung an dieBundID werden derzeit die beiden Bindings HTTP-REDIRECT und HTTP-POST unterstützt. Beide Bindings haben nachfolgend erwähnte Vor- und Nachteile im Praxisbetrieb, die im Rahmen der Architekturüberlegungen seitens der Bereitsteller von Drittanwendungen abzuwägen sind.

Das HTTP-REDIRECT-Binding ermöglicht das Übersenden des SAMLRequests als HTTP-GET-Parameter in der Anfrage-URL an den Identity Provider und unterliegt folglich den individuell regulierbaren Längenbegrenzungen aller zwischen Drittanwendung und Identity Provider befindlichen aktiven und passiven Netzwerkkomponenten (also Switches, Web-Application-Firewalls, Proxies, Web-Server, Personal Firewalls der Benutzer, etc.).

Bei Nutzung des HTTP-POST-Bindings wird der SAMLRequest innerhalb des HTTP-Body übermittelt, so dass aktive und passive Netzwerkkomponenten den Protokollablauf nicht in der gleichen Weise wie oben beschrieben, negativ beeinflussen können. Andererseits ist in diesem Zusammenhang die Nutzung der „Zurück“-Funktionalität bei einem gewollten Abbruch auf der Identity Provider-Seite browserabhängig beeinträchtigt. Die Folge kann eine Weiterleitungsschleife sein, die den Benutzer bei Abbruch des Loginvorgangs immer wieder auf den Identity Provider vorwärtsleitet. Dieses Problem zeigt sich generell bei Nutzung des SAML-HTTP-POST-Bindings und in kein Spezifikum der hier genutzten Produkte.

Tendenziell soll daher die Benutzung des HTTP-POST-Bindings bevorzugt werden.

8.2 Zurückleiten in die Drittanwendung bei Abbruch durch Benutzer

Aufgrund der Architektur der meisten Drittanwendungen bei Nutzung des HTTP-POST-Bindings, ist besonderes Augenmerk auf die Usability des Zurück-Buttons zu legen. Für eine reibungslose Zurückleitung in die Drittanwendung ist ein geordnetes Session-Handling der Drittanwendung gefordert, um ein zyklisches erneutes Absenden eines SAML-Requests und dessen Interpretation als Replay-Attacke zu vermeiden.

Umgesetzt werden kann ein solches Vorgehen z. B.: Durch Setzen und Prüfen eines zusätzlichen Cookies zum Zeitpunkt der Erstellung des SAML-Requests der Art:

```
if found_AlreadySentSAMLRequestCookie():
    invalidate_AlreadySentSAMLRequestCookie()
    forwardTo(previousStatus)
else:
```

8.3 Zurück zum Fachverfahren

Der "Zurück" Button in der ersten Anmeldeseite vom IDP dient zur Zurücknavigation zum aufrufenden Fachverfahren. Die sprachspezifische Zurücknavigation-URLs können durch optionale BackURL-Elemente in den Metadaten definiert werden. Die Entscheidung zwischen BackURL oder SAML-Response bleibt dem SP überlassen.

8.3.1 URL aus den Metadaten

Beispielhafter Auszug. akdb:BackURL kann gesetzt werden. Einträge pro Sprache sind optional. Falls nur einer vorhanden ist, wird dieser genommen. Falls die Sprache nicht gefunden wird, wird der erste Eintrag verwendet.

```
<md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
 <md:Extensions>
  <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
   <akdb:BackURL xmlns:akdb="https://www.akdb.de/idp/metadata/ui" xml:lang="de">https://www.example.com/de/back?a=1&amp;b=2&amp;error=user_cancelled&amp;lang=de</akdb:BackURL>
   <akdb:BackURL xmlns:akdb="https://www.akdb.de/idp/metadata/ui" xml:lang="en">https://www.example.com/en/back?a=1&amp;b=2&amp;error=user_cancelled&amp;lang=en</akdb:BackURL>
  </mdui:UIInfo>
```

Das Attribut xmlns:akdb="https://www.akdb.de/idp/metadata/ui" definiert den Namensraum für die Erweiterung von SAML-XML. Die URL "https://www.akdb.de/idp/metadata/ui" muss im gesamten Dokument unique sein. Zu beachten ist, dass nicht spezifiziert ist, dass diese URL auf irgendein bestimmtes Ziel zeigen muss. Das Präfix "akdb" wird benutzt um die Erweiterung-Elemente und Attribute in Metadaten eindeutig mit dem Namensraum zu verknüpfen. Diese Zeichenfolge (z. B. "akdb") ist frei wählbar, muss aber im gesamten Metadaten-Dokument eindeutig sein. Das Präfix muss NCName sein (siehe https://www.w3.org/TR/xml-names/#NT-NCName).

8.3.2 SAML-Response

Falls keine BackURL konfiguriert ist, erhält der SP beim Click auf zurück des Users eine SAML-Response, solange die Sitzung des Nutzers am IdP noch aktiv ist. Das Verhalten ist hierbei analog zu Elster oder eIDAS.

Auszug:

```
<saml2p:Response ID="_2aff60381b89b2b79d61df834af072e70810315a"
InResponseTo="ID_43fd6782-7ef0-429b-8e30-4549238154f3">
<saml2p:Status>
<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Requester">
<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:AuthnFailed"/>
</saml2p:StatusCode>
</saml2p:Status>
```

9 Konfiguration der Anfrage

Über eine Extension ist es möglich die Anfrage weiter anzupassen, als das mit Standard SAML2 Elementen möglich wäre. Der SAML-Request muss signiert sein, damit die Extension ausgewertet wird.

Die Hülle der Anfrage sieht folgendermaßen aus:

```
<akdb:AuthenticationRequest xmlns:akdb="https://www.akdb.de/request/2018/09" Version="1">
</akdb:AuthenticationRequest>
```

Achtung: `Version` ist ein Pflichtfeld. Aktuell wird Version 2 unterstützt. Die nachfolgenden Abschnitte beschreiben jeweils ein Kindelement der Extension.

Die Verwendung des AuthenticationRequests in Version 2 kann von den Betreibern eines Nutzerkontos als verpflichten eingestellt werden. Die Information ob und wann das stattfinden wird, wird durch die jeweilig Verantwortlichen kommuniziert. In der Regel geht das einher, mit der Forderung, den OrganizationDisplayName als verpflichtend zu übergeben.

Damit müssen im SAML-Request folgende Bedingungen verpflichtend erfüllt werden:

- Der Request muss signiert sein.
- OrganizationDisplayName muss übergeben werden.
- akdb:AuthenticationRequest in Version 2 muss verwendet werden
- akdb:RequestedAttributes müssen mindesten ein akdb:RequestedAttribute anfordern

Falls diese Bedingungen nicht erfüllt sind, antwortet der IDP direkt mit einer SAML-Response, da der Request nicht verarbeitet werden kann.

9.1 Einschränkung des Authentifizierungsverfahren

Zusätzlich zu Kapitel 5.1, können die Authentifizierungsverfahren auch folgendermaßen eingeschränkt werden. Die Smart-eID kann nicht als dediziertes Verfahren angefordert werden, da sie fachlich mit der eID gleichgestellt werden soll.

Es werden die aufgelisteten Authentifizierungsverfahren herangezogen, die kein <akdb:Enabled> definiert haben oder deren Wert true entspricht.

Version 2

```
<akdb:AuthnMethods>
    <akdb:Authega>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:Authega>
    <akdb:Benutzername>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:Benutzername>
    <akdb:eID>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:eID>
    <akdb:eIDAS>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:eIDAS>
    <akdb:Diia>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:Diia>
    <akdb:Elster>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:Elster>
    <akdb:FINK>
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:FINK>
</akdb:AuthnMethods>
```

Die unterstützten Werte entsprechen denen aus Kapitel 6.3.5 (AssertionProvedBy). Die temporäre Anmeldung ist kein eigenständiges Verfahren, weswegen dieser nicht über diese Liste abgewählt werden kann.

9.2 Anforderung von Pflichtattributen

Im SAML Request können Pflichtattribute angefordert werden. Die Implementierung dient grundsätzlich zur Gewährleistung der Datensparsamkeit, indem die Attribute spezifiziert werden müssen, deren Daten aus dem Nutzerkonto an den Aufrufer übertragen werden sollen. Die Verwendung von <akdb:RequestedAttributes> ist bei Nutzung des <akdb:AuthenticationRequest> **verpflichtend**. Es

werden nur die Attribute zurückgeliefert, die aufgelistet sind. Fehlende <akdb:RequestedAttributes> oder leere <akdb:RequestedAttribute>s führen dazu, dass die Extension im SAML-Request ignoriert wird. Mit RequiredAttribute können Attribute als verpflichtend gekennzeichnet werden. Dadurch wird garantiert, dass die Antwort (im Erfolgsfall) die verpflichtenden Attribute beinhaltet. Falls die erforderlichen Daten nicht vorliegen, wird eine SAML-Response als Fehler ohne Daten übertragen. RequiredAttribute ist per Default false.

Version 2

```
<akdb:RequestedAttributes>
    <akdb:RequestedAttribute Name="urn:oid:2.5.4.18" RequiredAttribute="false" />
    <akdb:RequestedAttribute Name="urn:oid:1.2.40.0.10.2.1.1.149" RequiredAttrib-
ute="true" />
    <akdb:RequestedAttribute Name="urn:oid:1.3.6.1.4.1.25484.494450.3"/>
</akdb:RequestedAttributes>
```

Das XML Attribut Name muss SAML2 Formal Name (URN-notiert) sein.

9.2.1 Technische Attribute vom Servicekonto und die Authentisierungsverfahren

Es können nachfolgende Attribute angefordert werden:

- bPK2 (urn:oid:1.3.6.1.4.1.25484.494450.3)
- legacyPostkorbHandle (urn:oid:2.5.4.18)

Werden ein, mehrere oder alle Attribute angefordert werden, werden die Authentisierungsverfahren ausgeblendet, welche die geforderten Attribute nicht erfüllen.

Sofern eine Drittanwendung beispielhaft ein bPK2 zwingend erfordert, kann die Drittanwendung im SAML-Request mindestens eines der Attribute anfordern, um das Ausblenden der temporären Anmeldung zu erreichen.

Bitte beachten Sie, dass sich die Daten, die ein Verfahren liefert, über die Zeit ändern können. FINK wurde zuerst ausgeblendet bei einer Anfrage nach bPK2, jedoch wurde das Verhalten über die Zeit geändert, da mittlerweile auch über den Portalverbund Kennzeichen übertragen werden, jedoch noch kein passendes Attribut um ein legacyPostkorbHandle zu unterstützen. Das Attribut bPK ist abgekündigt und wird bald nicht mehr unterstützt (siehe Kapitel zu bPK).

9.2.2 Validieren von RequestedAttributes im SAML Request

Die Requests mit der leeren Liste von RequestedAttributes oder mit falsch formatierten Attribut-Namen führt zum Ausschluss des <akdb:AuthenticationRequest>. Der IdP verhält sich so, als ob kein

<akdb:AuthenticationRequest> übermittelt wäre. Dieses Verhalten ändert sich, wenn V2 der Schnittstelle verpflichtend ist. Danach gelten diese leeren Anfragen als Fehler und der SAML-Request wird abgewiesen.

```
<akdb:RequestedAttributes>
</akdb:RequestedAttributes>
```

```
<akdb:RequestedAttributes>
    <akdb:RequestedAttribute Name="" />
</akdb:RequestedAttributes>
```

9.2.3 Validieren von RequestedAttributes vor dem Absenden

Falls die angeforderten Attribute vom Servicekonto nicht bereitgestellt werden können, wird im SAML Response mit SAML statuscode=urn:oasis:names:tc:SAML:2.0:status:RequestDenied geantwortet.

9.3 Änderung des Einleitungstexts

Der `Purpose` ist auf allen Seiten am IDP sichtbar. Nur gewisse HTML Elemente (u.a. `h1` `p` `a` `b` `i`) und Attribute (`href` auf `a`) sind erlaubt. Des weiteren nur https-URLs für href.

```
<akdb:DisplayInformation>
  <classic-ui:Version xmlns:classic-ui="https://www.akdb.de/request/2018/09/classic-ui/v1">
    <classic-ui:Purpose>
      <![CDATA[<h1>My HTML</h1>]]>
    </classic-ui:Purpose>
  </classic-ui:Version>
</akdb:DisplayInformation>
```

9.4 Übergabe von UI-Informationen (ab Release 6)

Purpose bleibt bestehen (siehe oben).

OrganizationDisplayName wird vor der Übermittlung der Daten des Benutzers vom IDP an den Onlinedienst angezeigt. Die Übergabe ist 12 Monate nach dem Rollout von Release 6 verpflichtend. Bei Werten mit mehr als 50 Zeichen behalten wir uns vor den Text im UI entsprechend zu kürzen. Aus Gründen der Abwärtskompatibilität wird im Übergangszeitrum bei fehlendem OrganizationDisplayName ein neutraler Wert angezeigt. Der Wert kann auch für Releases davor übergeben werden, hat dann aber keine Auswirkung. Zuvor wurden die Daten aus den SAML-Metadaten aus Organization/OrganizationDisplayName verwendet, weswegen der technische Begriff hier wieder aufgenommen

wurde. Fachlich wird der Wert auf https://id.bayernportal.de auch im Kontext der Anmeldeinformationen für den Benutzer „Anmelden im Online-Verfahren „${OrganizationDisplayName}““ verwendet (Stand 7. Juli 2023). Es besteht hier keine Anforderung den Wert der Organisation zu übermitteln, die den SAML-Request stellt, sondern es können fachlich hilfreiche Informationen übermittelt werden.

Lang kann verwendet werden, wenn im Onlinedienst bereits die gewünschte Sprache des Benutzers bekannt ist, damit dieser am SK nicht nochmals die Sprache wechseln muss. Valide Werte sind de, en, ru, uk. Der Wert ist optional und kann auch für Releases davor übergeben werden, hat dann aber keine Auswirkung. Der Default ist de.

BackURL wird für „zurück zum Onlinedienst“ verwendet. Die Übergabe ist 12 Monate nach dem Rollout von Release 6 verpflichtend. Bis dahin wird der vorhandene Weg verwendet (zurück mit SAML-Response ohne Werte oder in den Metadaten vorhanden URL). Die URL wird nicht validiert, sollte aber https verwenden und keine Möglichkeit bieten Zugriff auf Daten des Benutzers zu erlangen.

```
<akdb:DisplayInformation>
  <classic-ui:Version xmlns:classic-ui=”https://www.akdb.de/request/2018/09/classic-ui/v1”>
    <classic-ui:Purpose>
      <![CDATA[<h1>My HTML</h1>]]>
    </classic-ui:Purpose>
  </classic-ui:Version>
</akdb:DisplayInformation>
```

OnlineServiceId (ab Release 7) Im Schnittstellen-Request soll das Attribut Onlinedienst den Wert der BMI ID enthalten, die im Anbindungsprozess durch das BMI vergeben wurde. Damit soll erreicht werden, dass trotz Verwendung geteilter Länder-Zertifikate eine eindeutige Erkennung des Onlinedienstes möglich ist. (Stand Oktober 2023: Die OnlineServiceId findet aktuell keine fachliche Verwendung, bleibt aber hier weiter als Teil der Schnittstelle dokumentiert.)

```
 <akdb:DisplayInformation>
   <classic-ui:Version xmlns:classic-ui="https://www.akdb.de/request/2018/09/classic-ui/v1">
     <classic-ui:Purpose>
       <![CDATA[<h1>My HTML</h1>]]>
     </classic-ui:Purpose>
     <classic-ui:OrganizationDisplayName>
       <![CDATA[Meine Organisation>]]>
     </classic-ui:OrganizationDisplayName>
     <classic-ui:Lang>de</classic-ui:Lang>
     <classic-ui:BackURL>
       <![CDATA[https://example.com?a=1&b=2]]>
     </classic-ui:BackURL>
     <classic-ui:OnlineServiceId>
       <![CDATA[89479871264-DE]]>
```

```
    </classic-ui:OnlineServiceId>
  </classic-ui:Version>
</akdb:DisplayInformation>
```

9.5 Berechtigungszertifikat eines Bundeslandes

Der Service Provider kann damit andeuten, dass für eID und eIDAS das Berechtigungszertifikat des Bundeslandes verwendet wird. (Leitfaden für den Beantragungsprozess eines Länder-Berechtigungszertifikats siehe Dokument „BundID_eID_Berechtigungszertifikate_Juli_2023.pdf“.) Dafür müssen aber die organisatorischen Vorbereitungen getroffen sein, damit diese auch vom Betreiber des Servicekontos hinterlegt wurden. Alternativ kann das auch in den Metadaten des Service Providers hinterlegt werden, was dann aber kein setzen im SAML-Requests mehr ermöglicht. Für die Abkürzungen siehe auch https://www.destatis.de/DE/Methoden/abkuerzung-bundeslaender-DE-EN.html (DE ist kein gültiger Wert). Falls der Standard des Nutzerkontos gewünscht wird, muss der Eintrag weggelassen werden.

Mit Version 2 wird es als Kindelement von <akdb:eID> definiert.

Version 2

```
<akdb:AuthnMethods>
    <akdb:eID>
        <akdb:Berechtigungszertifikat Bundesland="BY" />
        <akdb:Enabled>true</akdb:Enabled>
    </akdb:eID>
    ...
</akdb:AuthnMethods>
```

Beispielhafter Auszug zur Konfiguration über Metadaten (relevanter Teil ist <akdb:Berechtigungszertifikat Bundesland="BY" xmlns:akdb="https://www.akdb.de/request/2018/09"/>). Das ist nur relevant, wenn ein Betreiber des Nutzerkontos auch unterschiedliche Berechtigungszertifikate anbietet.

Metadaten

```
<md:SPSSODescriptor WantAssertionsSigned="true" protocolSupportEnumeration="urn:oa-
sis:names:tc:SAML:2.0:protocol">
    <md:Extensions>
        <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
            <mdui:DisplayName xml:lang="de">Beispiel der Konfiguration des Berechtigungszer-
tifikats über Metadaten</mdui:DisplayName>
        </mdui:UIInfo>
        <akdb:Berechtigungszertifikat Bundesland="BY" xmlns:akdb="https://www.akdb.de/requ-
est/2018/09"/>
```

```
    </md:Extensions>
```

Falls ein Berechtigungszertifikat angefragt wird, das nicht hinterlegt ist, wird die komplett ignoriert (Stand Release 7).

9.6 Verwendete Fehlercodes am IDP bei unerwarteten SAML-Requests

Nicht alle Codes können zur gleichen Zeit auftreten und manche sind abhängig von der Konfiguration des jeweiligen Nutzerkontos. Der Code wird in der Regel in der Fehlerseite des NKs angezeigt und taucht in der URL als idp.code auf.

akdb-extension-missing
Anfrage mit fehlender AKDB-Extension.

akdb-extension-v1-deprecated
Anfrage mit veralteteter AKDB-Extension.

saml-request-not-signed
SAML-Request ist nicht signiert.

purpose-does-not-support-restricted-html
Purpose mit HTML, das nicht den Richtlinien entspricht.

missing-berca
Angefragtes Berechtigungszertifikat ist nicht hinterlegt.

requested-attributes-empty
Requested Attributes wurden nicht übergeben.

requested-attributes-missing-name
Requested Attributes ohne Name.

empty-authentication-methods
Leere Liste für Authentication Methods (keine Liste oder mind. ein Eintrag werden erwartet.

organization-display-name-missing
Name des Onlinedienstes fehlt.

9.7 Erweiterte Fehlermeldung im SAML-Response

Das Attribut EnableStatusDetail schaltet die erweiterte/detalierte Status-Meldung ein.

Das Setzen des Attributes.

```
<akdb:AuthenticationRequest xmlns:akdb="https://www.akdb.de/request/2018/09" Version="2" Ena-
bleStatusDetail="true">
	...
</akdb:AuthenticationRequest>
```

SAML-Response mit der detallierten Fehlermeldung

```
<saml2p:Response>
	...
	<saml2p:Status>
		<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Requester">
			<saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:RequestDe-
			nied"/>
		</saml2p:StatusCode>
		<saml2p:StatusMessage>security-msg</saml2p:StatusMessage>
		<saml2p:StatusDetail>
			<akdb:StatusDetail xmlns:akdb="https://www.akdb.de/request/2018/09">
				{
					"version": "1.0",
					"errors": [{
						"code": "IDP_REQUIRED_ATTRIBUTES_MISSING",
						"message":"Attribute
urn:oid:0.9.2342.19200300.100.1.20 is required but not available in Nutzerkonto. Attribute
urn:oid:0.9.2342.19200300.100.1.42 is required but not available in Nutzerkonto. Attribute
urn:oid:2.5.4.42 is required but not available in Nutzerkonto."
						}
					]
				}
			</akdb:StatusDetail>
		</saml2p:StatusDetail>
	</saml2p:Status>
</saml2p:Response>
```

10 Beispiele für SAML-Requests und SAML-Responses

10.1 Beispiele für SAML-Requests

```
<saml2p:AuthnRequest
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
AssertionConsumerServiceURL="https://samltool-ewg.pre.buergerserviceportal.de/saml/SSO"
Destination="https://pre-d-bayernid.freistaat.bayern/idp/profile/SAML2/POST/SSO"
ForceAuthn="true"
ID="8b3460e7-da7d-454b-80ca-4068b2237530"
IsPassive="false"
IssueInstant="2022-03-04T19:29:04.827Z"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Version="2.0">
<saml2:Issuer
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
https://samltool-ewg.pre.buergerserviceportal.de
</saml2:Issuer>
<saml2p:Extensions>
<akdb:AuthenticationRequest
xmlns:akdb="https://www.akdb.de/request/2018/09"
Version="1">
<akdb:AllowedMethods>
<akdb:AuthnMethod>
FINK
</akdb:AuthnMethod>
<akdb:AuthnMethod>
eID
</akdb:AuthnMethod>
<akdb:AuthnMethod>
Benutzername
</akdb:AuthnMethod>
</akdb:AllowedMethods>
<akdb:RequestedAttributes>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.18"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.3.6.1.4.1.25484.494450.3"
/>
</akdb:RequestedAttributes>
<akdb:Berechtigungszertifikat
Bundesland="BY"
/>
</akdb:AuthenticationRequest>
</saml2p:Extensions>
<saml2p:RequestedAuthnContext
Comparison="minimum">
<saml2:AuthnContextClassRef
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
STORK-QAA-Level-3
</saml2:AuthnContextClassRef>
</saml2p:RequestedAuthnContext>
</saml2p:AuthnRequest>


<saml2p:AuthnRequest
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
AssertionConsumerServiceURL="https://samltool-ewg.pre.buergerserviceportal.de/saml/SSO"
Destination="https://pre-d-bayernid.freistaat.bayern/idp/profile/SAML2/POST/SSO"
ForceAuthn="true"
ID="e0dc8abd-d5cb-4767-8fa7-5c581c9f8aa7"
IsPassive="false"
IssueInstant="2022-03-04T19:30:38.033Z"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Version="2.0">
<saml2:Issuer
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
https://samltool-ewg.pre.buergerserviceportal.de
</saml2:Issuer>
<saml2p:Extensions>
```

```
<akdb:AuthenticationRequest
xmlns:akdb="https://www.akdb.de/request/2018/09"
Version="2">
<akdb:AuthnMethods>
<akdb:eID>
<akdb:Berechtigungszertifikat
Bundesland="BY"
/>
<akdb:Enabled>
true
</akdb:Enabled>
</akdb:eID>
<akdb:FINK>
<akdb:Enabled>
true
</akdb:Enabled>
</akdb:FINK>
</akdb:AuthnMethods>
<akdb:RequestedAttributes>
<akdb:RequestedAttribute
Name="urn:oid:1.3.6.1.4.1.33592.1.3.5"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.3.6.1.4.1.25484.494450.3"
RequiredAttribute="true"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.3.6.1.5.5.7.9.2"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.16"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.17"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.2.40.0.10.2.1.1.225599"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.2.40.0.10.2.1.1.225566"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:1.2.40.0.10.2.1.1.225577"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.18"
RequiredAttribute="true"
/>
<akdb:RequestedAttribute
Name="urn:oid:0.9.2342.19200300.100.1.40"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.7"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.42"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid: 1.2.40.0.10.2.1.1.261.94"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:2.5.4.4"
RequiredAttribute="false"
```

```
/>
<akdb:RequestedAttribute
Name="urn:oid:1.2.40.0.10.2.1.1.55"
RequiredAttribute="false"
/>
<akdb:RequestedAttribute
Name="urn:oid:0.9.2342.19200300.100.1.3"
RequiredAttribute="false"
/>
</akdb:RequestedAttributes>
</akdb:AuthenticationRequest>
</saml2p:Extensions>
<saml2p:RequestedAuthnContext
Comparison="minimum">
<saml2:AuthnContextClassRef
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
STORK-QAA-Level-3
</saml2:AuthnContextClassRef>
</saml2p:RequestedAuthnContext>
</saml2p:AuthnRequest>
```

10.2 Beispiele für SAML-Responses

siehe Anhang 1

11 Postfach-Nachricht an die BundID

Hinweis zur Abkündigung der SOAP-Schnittstelle:

Die SOAP-Schnittstelle der BundID wird zukünftig durch die Schnittstelle für das zentrale Bürgerpostfach (ZBP) ersetzt, allerdings noch bis mindestens 31.12.2024 unterstützt.

Mit Release 8.0.0.0 ist die neue Schnittstelle zum Postfach auf Basis von REST (ZBP) verfügbar. Verwenden Sie hierzu bitte die entsprechende Dokumentation.

Nachfolgende Schnittstellendokumentation ist daher zukünftig nicht mehr zu verwenden.

Im vorliegenden Dokument wird die Schnittstelle für den Nachrichtenversand an das Postfach in der BundID beschrieben.

Diese Schnittstelle bietet externen Online-Leistungen und Fachverfahren die Möglichkeit, Nachrichten in das Postfach eines bestimmten Nutzenden abzulegen.

11.1 Topologie

Direkte Verbindung

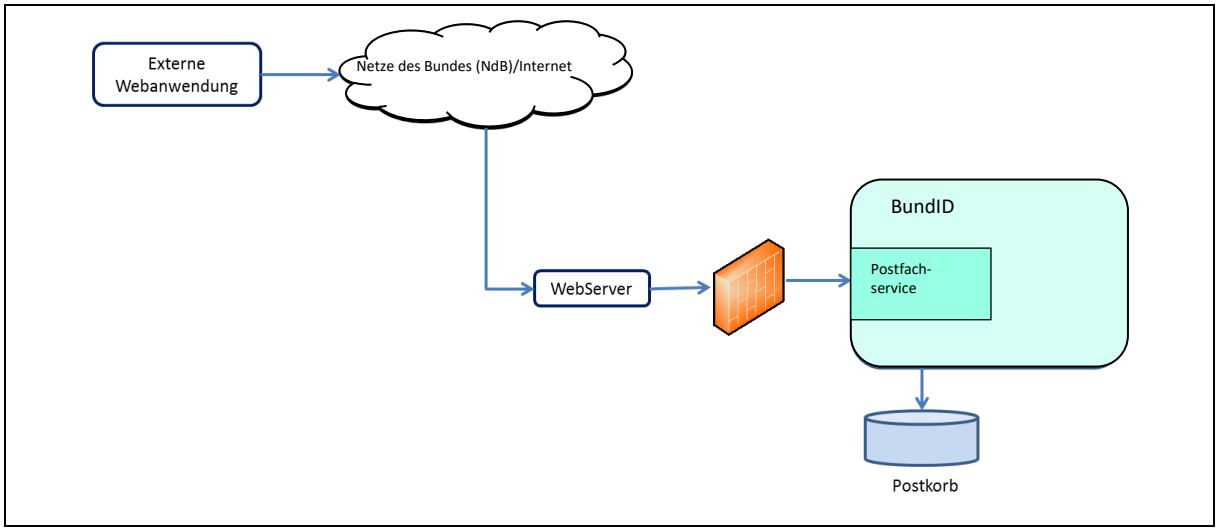


Abbildung 5: Schnittstelle Nachrichtenversand

11.2 Der Nachrichtentransport erfolgt zwischen der externen Webanwendung und dem Bundesportal per HTTPS-Soap. Die Verbindung muss über die Netze des Bundes (NdB) erfolgen. SOAP-Web-Service

Die Übergabe der Postfachnachrichten erfolgt über eine synchrone SOAP-Webservice-Schnittstelle per HTTPS (Vgl. Kapitel 14 Web-Service-Schnittstelle).

Die Postfachnachricht erfolgt im XML-Format. Als Antwort wird eine XML-Antwort mit einer speziellen „Postfach-Empfangs-Quittung“ geliefert.

Die Antwort wird unmittelbar nach Übernahme der Postfachnachricht zurückgeschickt; die aufrufende Anwendung kann auf die Antwort warten und entsprechend darauf reagieren.

Der Eingang ist in Kapitel 14 Web-Service-Schnittstelle beschrieben.

12 Adressierung

Die Postfach-Nachricht der BundID ist eine spezielle Form der allgemeinen und universellen Nachricht in der BundID, die prinzipiell beliebige Informationen vom oder zur BundID übertragen.

Die Postfach-Nachrichten im Kontext der BundID kennen nur eine Richtung, von der externen Webanwendung zur BundID.

Die allgemeine BSP-Nachricht der BundID sieht theoretisch unterschiedliche Kriterien für die Adressierung des Empfängers vor, wie z. B. Adressierung über einen amtlichen Gemeindeschlüssel (AGS), über eine Mandanten-ID, über Namen etc.

Für die Postfach-Nachricht im Kontext der BundID ist hingegen nur die Adressierung über das *Postkorb-Handle* vorgesehen, in der das Postfach der Nutzenden eindeutig identifiziert wird.

Jeder registrierte Nutzende hat einen einmaligen und eindeutigen Postkorb-Handle, welcher sich auf das persönliche Postfach bezieht.

Fiktives Beispiel für ein Postkorb-Handle:

S0tU_AL6FOnJm9IfIKkdq_1NqqWPLYOl1llptHbbes4

13 Das Format der Postfach-Nachricht

13.1 Nachrichtenaufbau informativ

Die Postfachnachricht ist eine Nachricht im XML-Format, deren Aufbau über ein XML/XSD-Schema (bspnachricht.xsd) definiert ist. Eine Nachricht besteht aus einem Nachrichtenkopf und dem Nachrichteninhalt:

- Der Nachrichtenkopf enthält technische Informationen sowie Absender und Empfänger-Daten.
- Der Nachrichteninhalt enthält die eigentliche Nachricht samt Anhängen, sowie zusätzliche fachliche Attribute zur Nachricht.

Das Schema spezifiziert allgemeine Nachrichten vom und zur BundID; es gilt aber insbesondere auch für Postfach-Nachrichten. Hierbei sind jedoch semantische Einschränkungen zu berücksichtigen, die nachfolgend näher erläutert werden.

Die für Postfach-Nachrichten zwingend erforderlichen Felder sind im Folgenden als **Pflichtfelder (***)** in Fettdruck gekennzeichnet.

Diese Felder müssen immer gefüllt sein, auch wenn sie im zugehörigen XML-Schema, welches prinzipiell auch alternative Adressierungsmöglichkeiten vorsieht, als optional gekennzeichnet sind.

Den Postfachnachrichten können mehrere Anhänge beigefügt werden. Die Limitierung liegt derzeit bei maximal 5 Anhängen mit je 2 Mbyte Größe.

13.1.1 Nachrichten-Kopf

- **NachrichtenID (***)**

  technisch, beliebig, GUID.
- **Erstellungszeitpunkt (***)**

  Datum, Uhrzeit
- **Absender (***)**
  - Postfach-ID (Postkorb-Handle) (<PostkorbId>)
  - Verfahren
  - **Dienst (***)** (erscheint in der Postfach-Ansicht des Bürgers am Portal)
  - **Mandant (***)** (erscheint als „Absender“ in der Postfach-Ansicht des Bürgers am Portal)
  - Gemeindeschlüssel
  - Name

    - Anschrift
    - Email-Adresse
    - Telefon
    - Hyperlink

- **Empfänger (***)**
    - **Postfach-ID (Postkorb-Handle) (<PostkorbId>) (***)**
    - Verfahren
    - Dienst
    - Absender
    - Gemeindeschlüssel
    - Name
    - Anschrift
- Antwort Auf / Weiterleitung Zu
- Lesebestätigung Antwortadresse

Erläuterungen:

- Die **Nachrichten-ID** sollte eindeutig pro Nachricht sein, sie hat nur informativen Charakter und wird vorerst nur für Debug-Zwecke oder Fehleranalysen benötigt.
- Der **Erstellungszeitpunkt** wird ebenfalls nur für Debug-Zwecke oder Fehleranalysen benötigt.
- Der **Dienst**-Name muss angegeben werden; er erscheint in der Postfachansicht der BundID in der Zeile „OnlineDienst“.
- Der **Mandant**-Name muss angegeben werden, er erscheint 1:1 in der Postfachansicht der BundID in der Spalte „Absender“. Der Absender kann mit einem Hyperlink zu dem gewünschten Ziel hinterlegt werden.

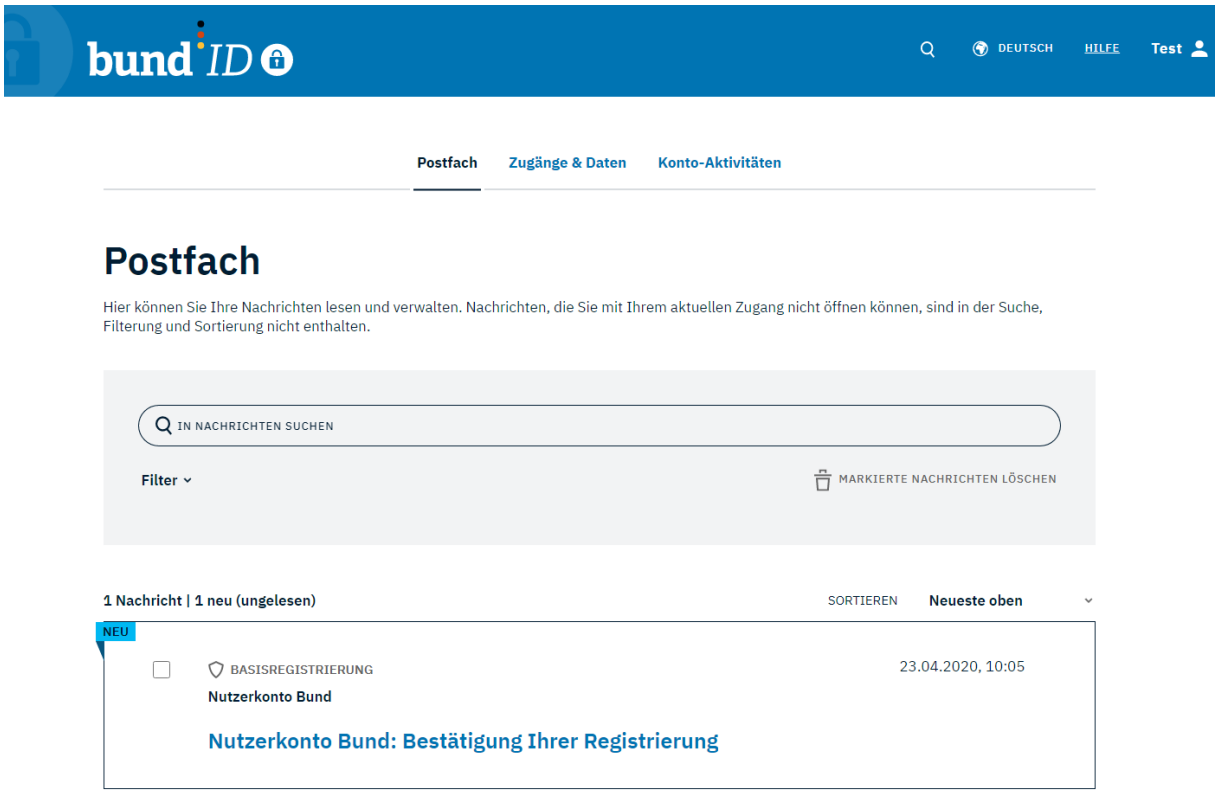


Abbildung 6: Postfach

- Die **Postfach-ID** (XML-Elementname: <PostkorbId>) (bzw. der Postkorb-Handle) des Empfängers muss mit einem existierenden Postfach-Handle besetzt sein. Der Postfach-Handle wird der externen Webanwendung im Zuge der Anmeldung an der Identity-Federation der BundID nach erfolgreicher Authentisierung übermittelt.
- Die optionale Nachrichten-Kopf-Komponente „Antwort Auf / Weiterleitung Zu“ ist zum Referenzieren zurückliegender Nachrichten gedacht. Sie wird vorerst nicht unterstützt.
- Die optionale Nachrichten-Kopf-Komponente „Lesebestätigung Antwortadresse“ (im Code: lesebestaetigungAntwortAdresse) ist für die Übermittlung einer E-Mailadresse im gültigen Format bestimmt. Ruft ein Nutzer die Nachricht in seinem Postfach auf, wird eine Nachricht an diese E-Mailadresse übermittelt. In der Nachricht wird der Betreff der Nachricht angegeben, der unter Nachrichteninhalt Betreff übergeben wird.

13.1.2 Nachrichten-Inhalt

- **Betreff (***)**
- Kategorie (Statusinformation, allgemeine Information, Information von der Kommune,…)
- Stork-QAA-Level (Nachricht darf nur nach entsprechender Authentisierung gelesen werden)

- Vorgang
  - VorgangsName
  - VorgangsId
  - Status (empfangen, aufgenommen, in Arbeit, Rückfrage, fertig, …)
- **Nachrichtentext (<*FreiText*>) (***)**
  - **encoding: HTML/ASCII (***)**
  - **Text (<*Text*>) (***)**
- Anhänge (0 .. n)
  - Inhalt (base64) (***) (Pflichtfeld nur falls
  - Filetype (text, html, jpeg, pdf, …) (***) ein Anhang vorhanden)
  - Dateiname

Erläuterungen:

- **Betreff** und **Nachrichtentext** sind zwingend erforderlich.
- Über den Stork-QAA-Level kann eine Nachricht dahingehend geschützt werden, dass sie nur dann sichtbar ist, wenn vorher eine erfolgreiche Authentisierung mit dem entsprechenden (Nicht-)Vertrauensniveau erfolgt ist.

Technische Bezeichnung	**Bezeichnung nach TR-03160-1 bzw. Beschluss der Projektgruppe eID Strategie von Juli 2021**
STORK-QAA-Level-1	Basisregistrierung
STORK-QAA-Level-2	Normal
STORK-QAA-Level-3	Substantiell
STORK-QAA-Level-4	Hoch

- Größe, Anzahl sowie zulässige Inhaltstypen für Nachrichteninhalte und Anhänge können durch technische Gegebenheiten eingeschränkt sein. Gegebenenfalls wird eine eingehende Postfachnachricht aus diesen Gründen abgelehnt.
- Derzeit werden technisch maximal 20 Anhänge pro Nachricht mit je 50 Mbyte Maximalgröße pro Anhang akzeptiert. Die BundID erlaubt derzeit maximal 5 Anhänge mit je maximal 2 Mbyte.
- Eine Liste der möglichen Anhangtypen ist in Tabelle 9005 in Abschnitt 13.2.6 Die Schlüsseltabellen, festgelegt. Wenn ein Dateiname mitgeliefert wird, muss dessen Dateinamenendung zum angegebenen Filetype korrespondieren.
- Bitte achten Sie bei der Verwendung von Dateinamen für Anhänge auf die Dateisystemanforderungen der verschiedenen Betriebssysteme! Theoretisch kann jeder Dateiname für einen

Anhang verwendet werden, praktisch lassen sich Anhänge mit inkompatiblen Benennungen vom Benutzer nicht öffnen oder herunterladen. Die Schnittstelle nimmt keine Prüfung diesbezüglich vor.

- Die aufgeführten MIME-Types (Tabelle 9005) sind die technisch verwendbaren MIME-Types. Diese können je Installation vom Betreiber eingeschränkt werden. Bitte erfragen sie die erlaubten MIME-Types vom Betreiber des Nutzerkontos, an welches sie Nachrichten senden möchten.

13.1.3 Validierung Nachrichten-Inhalt

Die Postkorbempfangsschnittstelle weist Nachrichten zurück, die unerlaubtes HTML enthalten. Per Setting htmlAllowedTags kann die Liste der zulässigen Html-Tags festgelegt werden.

Empfohlen werden folgende Tags (Setting htmlAllowedTags): b,i,u,strong,h1,h2,h3,h4,ul,ol,li,a,img,p,button,br,em

Der Sender erhält Status-Code 31 "Unerlaubter Nachrichteninhalte" der ihn darauf hinweist, dass er nicht unterstütze HTML-Elemente verwendet. Die unzulässigen Elemente werden im Einzelnen aufgelistet.

In der Nachricht darf nur https:// verwendet werden kein http://

Validiert werden 4 xml Elemente:

```
<BspNachricht xmlns="http://www.akdb.de/egov/bsp/nachrichten" version="1.2" fassung="2018-04-01">
    <NachrichtenKopf>
    ....
        <Absender>
          ......
        <Mandant>Ingolstadt</Mandant>                                    html-whitelist-valedierung 1
        ...
            <Hyperlink>http://www.akdb.de</Hyperlink>                    html-whitelist-valedierung 2
        </Absender>
       .....
    </NachrichtenKopf>
    <NachrichtenInhalt>
        <Betreff>Nachrichtenbetreff Fri Mar 25 05:44:52 GMT 2022</Betreff>        html-valedierung 3
     ......
        <FreiText>
            ....
            <Text>Nachrichtentext Fri Mar 25 05:44:52 GMT 2022</Text>   html-whitelist-valedierung 4
        </FreiText>
    </NachrichtenInhalt>
```

```
</BspNachricht>
```

13.2 XML-Nachrichtenschema für eine BSP-Nachricht

Das XML-Nachrichtenschema ist als XSD-Schema definiert.

Die aktuelle Version der Schemadatei lautet bspnachrichten-2.11.xsd

13.2.1 Siehe Anhang 14.6.2 Schemadatei - bspnachrichten-2.13.xsdBSP-Nachricht

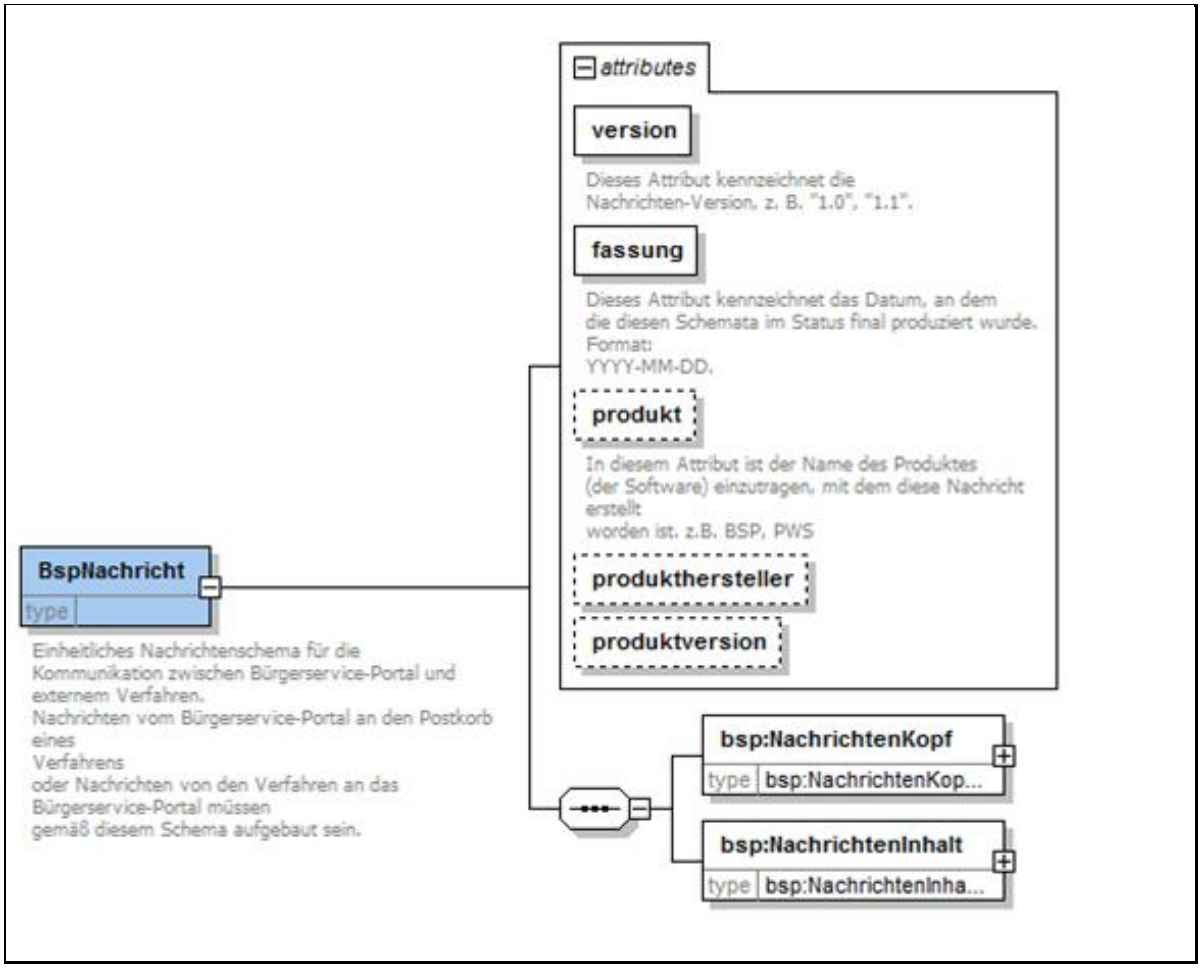


Abbildung 7: Schemadatei – bspnachrichten-2.11.xsdBSP-Nachricht

13.2.2 BSP-Nachrichtenkopf

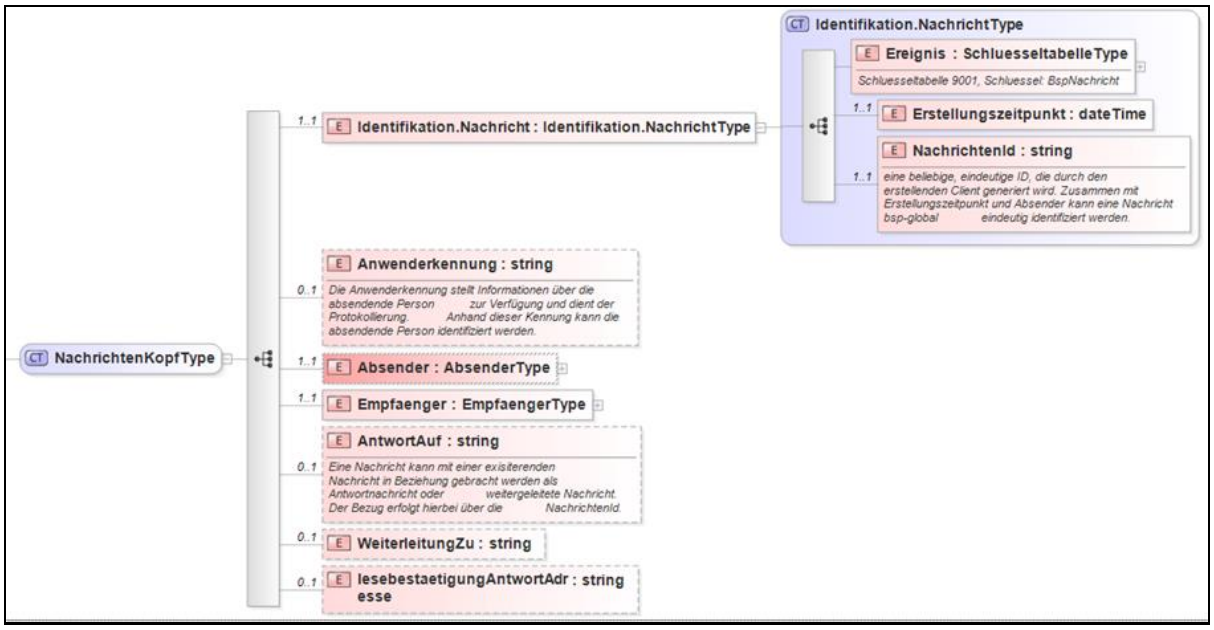
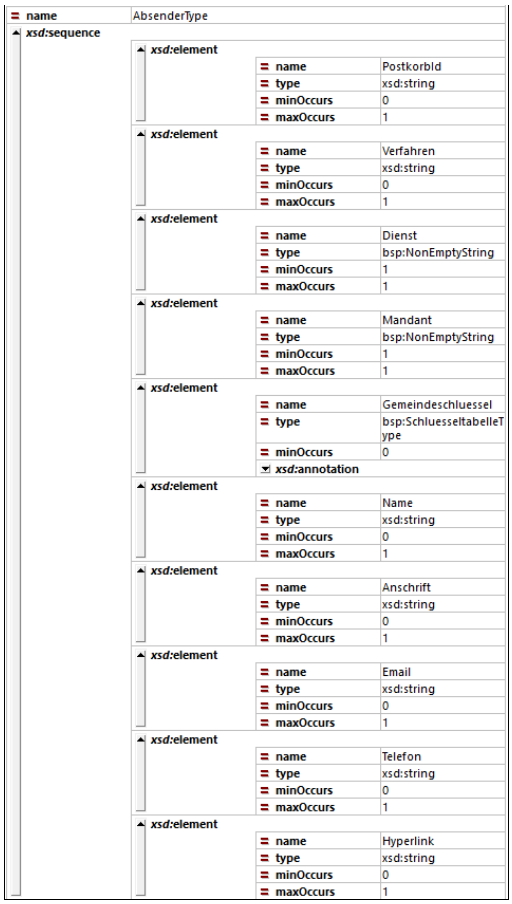


Abbildung 8: BSP-Nachrichtenkopf

13.2.3 Absender und Empfänger



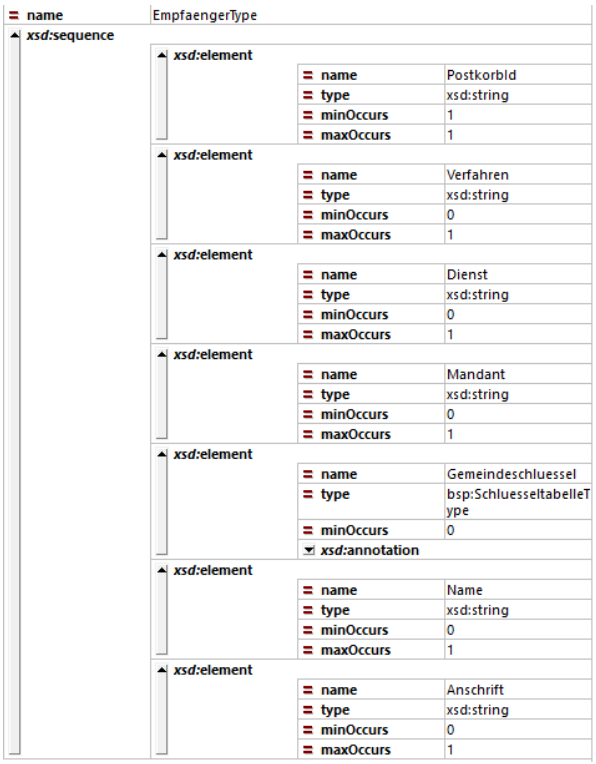


Abbildung 9: Absender und Empfänger

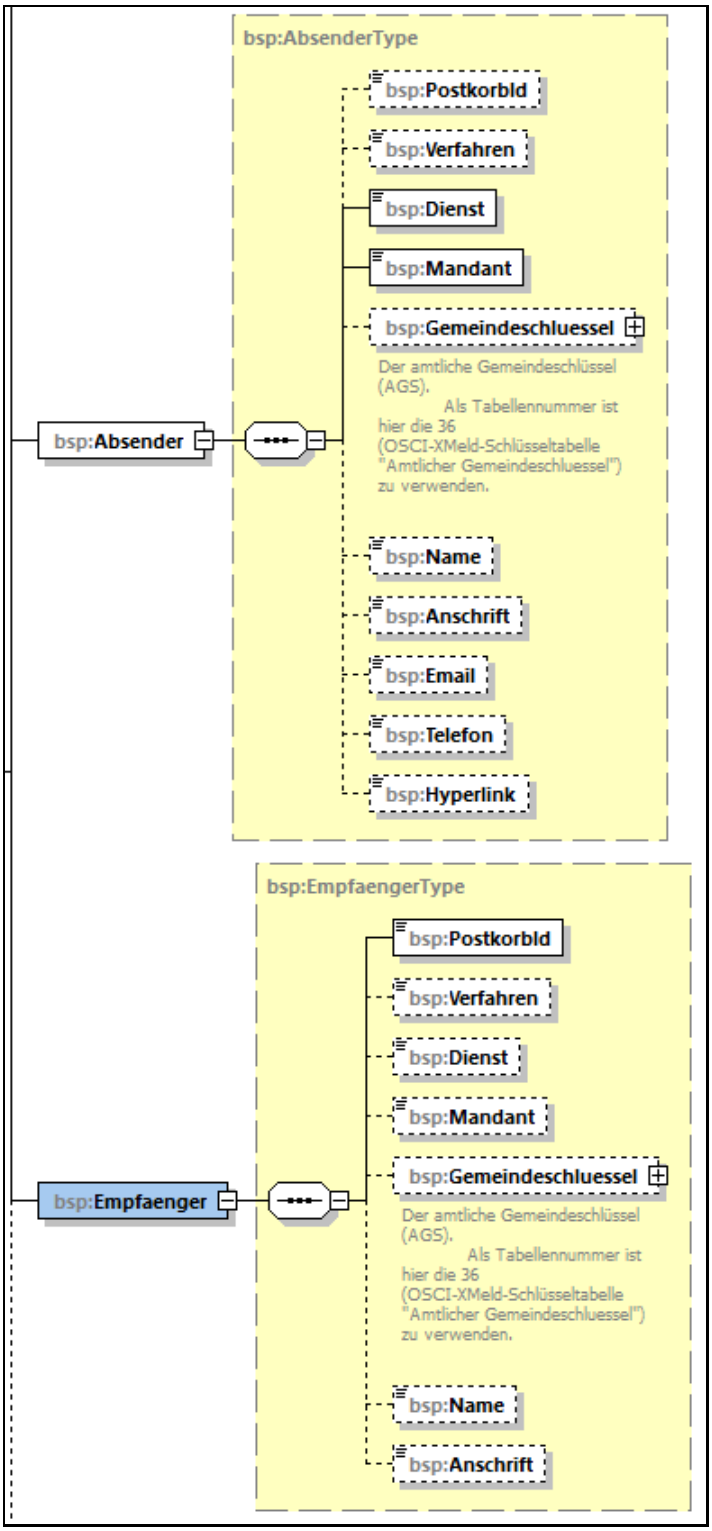


Abbildung 10: Absender und Empfänger

13.2.4 BSP-Nachrichteninhalt

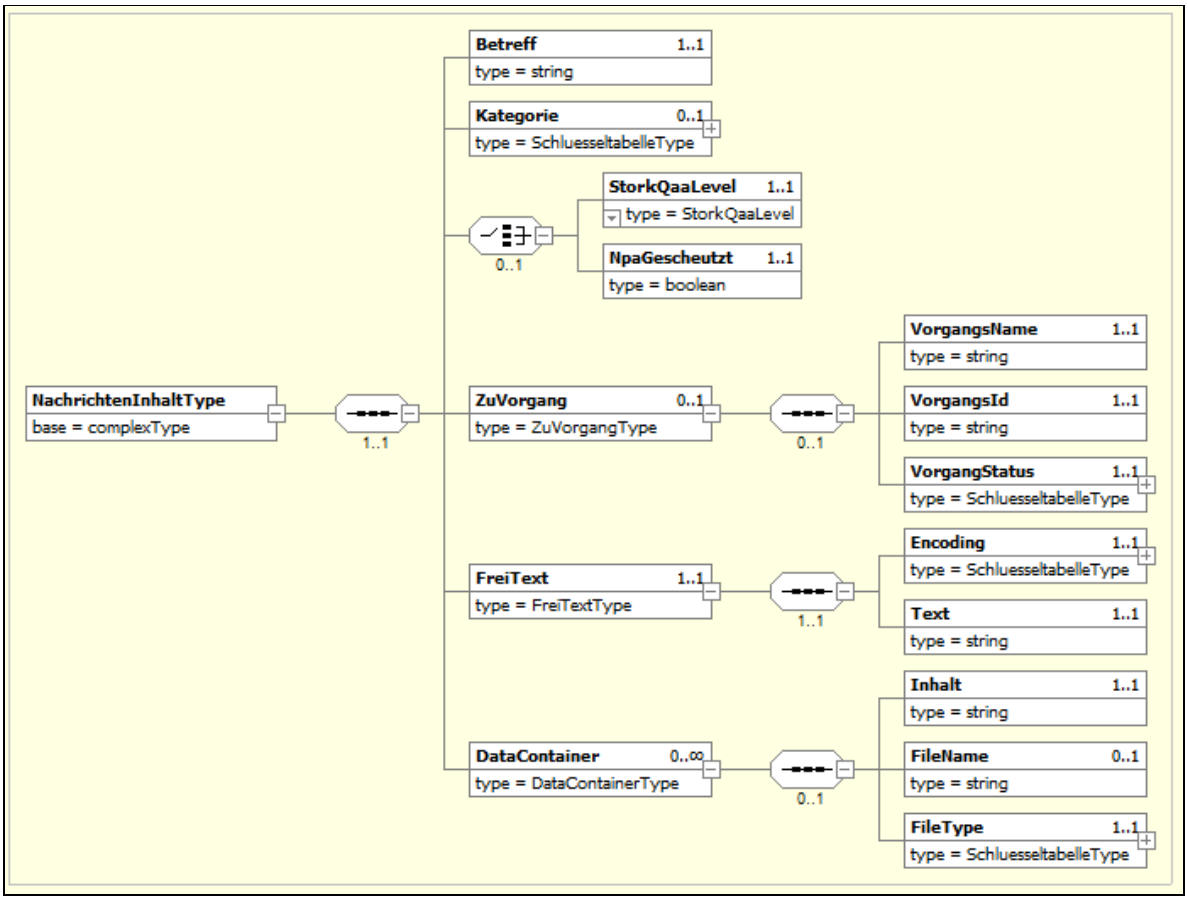


Abbildung 11: BSP-Nachrichteninhalt

13.2.5 BSP-Quittung

Der Ergebnisstatus gibt Auskunft über die Fehlerursache bei der Ablehnung einer Nachricht. Die möglichen Fehlersituationen sind in Schlüsseltabelle 9006 festgelegt.

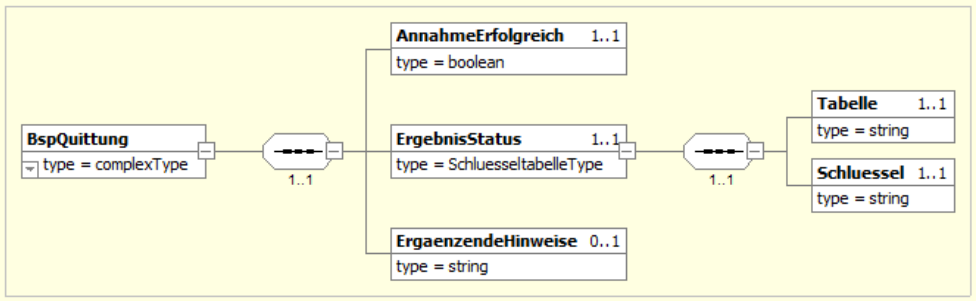


Abbildung 12: BSP-Quittung

Hinweis zu § 9 OZG (i. d. F. v. 03.12.2020):

Die BSP-Quittung wird nach der fachlichen Verarbeitung der Nachricht im Nutzerkonto synchron quittiert. Daher kann die Quittierung nach Tabelle 9006 als rechtssichere Annahme und Zustellung der Nachricht an das angesprochene Postfach gewertet werden.

13.2.6 Die Schlüsseltabellen

Die folgenden Schlüsseltabellen legen die zulässigen Werte für die jeweiligen Schlüsselfelder (z.B. bspEreignis im BSP-Nachrichtenkopf) in den Postfachnachrichten fest.

Diese Tabellen werden als Code-Listen gemeinsam mit dem XSD-Nachrichtenschema veröffentlich.

Tabelle Nr.	**9001**
Name	**Ereignisse**
Beschreibung	**Liste der möglichen Ereignisse**
Schlüssel	**Wert**
BSP	Nachricht vom oder zur BundID

Tabelle Nr.	**9002**
Name	**Kategorie**
Beschreibung	**Nachrichtenkategorien**
Schlüssel	**Wert**
KAT_STATUS	Statusmeldung
KAT_INFOBSP	Information von der BundID
KAT_INOKOMMUNE	Information von der Kommune

Tabelle Nr.	**9003**
Name	**VorgangStatus**
Beschreibung	**Vorgangsstatus**
Schlüssel	**Wert**
ST_ERHALTEN	Nachricht wurde erhalten
ST_GELESEN	Nachricht wurde gelesen
ST_IN_ARBEIT	Nachricht wird bearbeitet
ST_FERTIG	Nachricht fertig bearbeitet

Tabelle Nr.	**9004**
Name	**TextEncoding**
Beschreibung	**Codierungsvarianten für Freitexte**
Schlüssel	**Wert**
text/plain	Einfacher ASCII-Text
text/html	Der Inhalt wird im HTML-Format übermittelt.
text/rtf	Der Inhalt wird im RTF-Format übermittelt.
text/xml	Der Inhalt wird im XML-Format übermittelt.

Tabelle Nr.	**9005**
Name	**MIMETypes**
Beschreibung	**Codierungsvarianten für MIME-Anhänge**
Schlüssel	**Wert**
text/plain	Der Anhang wird als einfacher Text übermittelt (Dateinamenendung: „.txt“)
text/html	Der Anhang wird im HTML-Format übermittelt. (Dateinamenendung: „.htm, .html, , .shmtl“)
text/rtf	Der Anhang wird im RTF-Format übermittelt. (Dateienamenendung: „.rtf“
text/calendar	Der Anhang wird im ICS-Format übermittelt (Dateienamenendung: „.ics“)
text/comma-separated-values	Der Anhang wird im CSV-Format übermittelt (Dateienamenendung: „.csv“)
image/jpeg	Der Anhang wird als Bild im JPEG-Format übermittelt. (Dateinamenendung: „.jpg", ".jpe", ".jpeg", ".jfif")
image/gif	Der Anhang wird als Bild im GIF-Format übermittelt. (Dateinamenendung: „.gif“)
image/png	DerAnhang wird als Bild im PNG-Format übermittelt (Dateinamenendung: „.png“)
image/tiff	Der Anhang wird als Bild im TIF-Format übermittelt (Dateinamenendung: „.tiff“, ".tif")
image/bmp	Der Anhang wird als Bild im BMP-Format übermittelt (Dateinamenendung: „.bmp“)
image/svg+xml	Der Anhang wird als Bild im SVG-Format übermittelt (Dateinamenendung: „.svg“)
application/pdf	Der Anhang wird im PDF-Format übermittelt (Dateinamenendung: „.pdf“
application/acad	Der Anhang wird im DWG-Format übermittelt (Dateinamenendung: „.dwg“)
application/dxf	Der Anhang wird im DXF-Format übermittelt (Dateinamenendung: „.dxf“)
application/gzip	Der Anhang wird als Archiv im GZIP-Format übermittelt (Dateinamenendung: „.gz“)
application/octet-stream	Der Anhang wird als Byte-Stream übermittelt (Dateinamenendung beliebig)
audio/mp3	Der Anhang wird als Audio im MP3-Format übermittelt. (Dateinamenendung: „.mp3“)
audio/wav	Der Anhang wird als Audio im WAV-Format übermittelt. (Dateinamenendung: „.wav“)

video/mp4	Der Anhang wird als Video im MP4-Format übermittelt. (Dateinamenendung: „.mp4“)
video/mpeg	Der Anhang wird als Video im MPEG-Format übermittelt. (Dateinamenendung: „.mpeg“)

Tabelle Nr.	**9006**
Name	**Ergebnisstatus**
Beschreibung	**Ergebnisstatus des Empfangs einer Nachrichte**
Schlüssel	**Wert**
0	Nachricht wurde erfolgreich übernommen
20	Fehler im BSP-Nachrichtenschema
30	Ungültiger Postfach-Handle
31	Unzulässiger Nachrichteninhalt
32	Unzulässiger Nachrichtenanhang
99	Sonstiger technischer Fehler

13.2.7 Beispielnachricht

Beispielnachricht

```
<?xml version="1.0" encoding="utf-8"?>

<bsp:BspNachricht xmlns:bsp=http://www.akdb.de/egov/bsp/nachrichten
    xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
    xsi:schemaLocation=http://www.akdb.de/egov/bsp/nachrichten
    version="1.3" fassung="2018-11-01"
    produkt="PWS" produkthersteller="AKDB" produktversion="4.11" />

    <NachrichtenKopf>
        <Identifikation.Nachricht>
            <Ereignis>
                <Tabelle>9001</Tabelle>
                <Schluessel>BSP</Schluessel>
            </Ereignis>
            <Erstellungszeitpunkt>2013-05-11T09:30:47.0Z</Erstellungszeitpunkt>
            <NachrichtenId> XVdggWV112311HH </NachrichtenId>
        </Identifikation.Nachricht>

        <Absender>
            <Verfahren>PWS</Verfahren>
            <Dienst>Gehaltsnachweis</Dienst>
            <Mandant>Ingolstadt</Mandant>
            <Gemeindeschluessel>
                <Tabelle>36</Tabelle>
                <Schluessel>09161000</Schluessel>
            </Gemeindeschluessel>
            <Name>Paula Panther</Name>
            <Anschrift>Stadtverwaltung      In-
golstadt, Hauptplatz 12, 83231 Ingolstadt</An-
schrift>
```

```
            <Email>paula.pather@stadt-ingolstadt.de</Email>
            <Telefon>08372-123123</Telefon>
            <Hyperlink>http://www.ingolstadt.de</Hyperlink>
        </Absender>

        <Empfaenger>
            <PostkorbId>8911287002911</PostkorbId>
        </Empfaenger>
        <AntwortAuf></AntwortAuf>
        <WeiterleitungZu></WeiterleitungZu>
        <lesebestaetigungAntwortAdresse>lesebestaetigung@stadt-in-
golstadt.de</lesebestaetigungAntwortAdresse>
    </NachrichtenKopf>

    <NachrichtenInhalt>
        <Betreff>Testnachricht </Betreff>

        <StorkQaaLevel>STORK-QAA-Level-1</StorkQaaLevel>

        <FreiText>
            <Encoding>
                <Tabelle>9004</Tabelle>
                <Schluessel>text/plain</Schluessel>
            </Encoding>
            <Text>Sehr geehrter Herr Mustermann,\n\n
            Zur vollständigen Bearbeitung Ihres Antrages......
            </Text>
        </FreiText>

        <DataContainer>
            <Inhalt>fzewbsd63hsdsj3adh3413112dhs.....</Inhalt>
            <FileName>Bild1.pdf</FileName>
```

```
            <FileType>
                <Tabelle>9005</Tabelle>
                <Schluessel>application/pdf</Schluessel>
            </FileType >
        </DataContainer>
    </NachrichtenInhalt>
</bsp:BspNachricht
```

14 Web-Service-Schnittstelle

Der Postfach-Service wird über eine Webservice-Schnittstelle mit folgender WSDL angesprochen. Die WSDL ist auch per URL Aufruf einsehbar „…/bspx-postkorb-okkomm-ws/bspservices/postkorb-komm.wsdl“; bspw. https://int.id.bund.de/bspx-postkorb-okkomm-ws/bspservices/postkorb-komm.wsdl :

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="PostkorbKommService" xmlns="http://schemas.xmlsoap.org/wsdl/"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:bsp="http://akdb.de/portal/gehaltsabrechnungen-bspnachricht"
 xmlns:tns="urn:akdb:bsp:postkorb:komm:webservice" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="urn:akdb:bsp:postkorb:komm:webservice">
 <types>

  <xs:schema targetNamespace='urn:akdb:bsp:postkorb:komm:webservice'
   version='1.1' xmlns:tns='urn:akdb:bsp:postkorb:komm:webservice'
   xmlns:xs='http://www.w3.org/2001/XMLSchema'>

   <xs:element name='sendBspNachricht' type='tns:sendBspNachricht' />
   <xs:complexType name='sendBspNachricht'>
    <xs:sequence>
     <xs:element minOccurs='1' name='okKommBspNachrichtInput' type='xs:base64Binary' />
    </xs:sequence>
   </xs:complexType>

   <xs:element name='sendBspNachrichtOutput' type='tns:sendBspNachrichtOutput' />
   <xs:complexType name='sendBspNachrichtOutput'>
    <xs:sequence>
     <xs:element minOccurs='1' name='okKommBspNachrichtOutput' type='xs:base64Binary' />
    </xs:sequence>
   </xs:complexType>
```

```
   <xs:element name='sendBspNachrichtNative' type='tns:sendBspNachrichtNative' />
   <xs:complexType name='sendBspNachrichtNative'>
    <xs:sequence>
     <xs:element minOccurs='1' name='bspNachricht' type='xs:string' />
    </xs:sequence>
   </xs:complexType>

   <xs:element name='sendBspNachrichtNativeOutput' type='tns:sendBspNachrichtNativeOutput' />
   <xs:complexType name='sendBspNachrichtNativeOutput'>
    <xs:sequence>
     <xs:element minOccurs='1' name='bspQuittung' type='xs:string' />
    </xs:sequence>
   </xs:complexType>
  </xs:schema>
</types>

<message name='postkorbKommService_sendBspNachrichtInput'>
  <part element='tns:sendBspNachricht' name='okKommBspNachrichtInput'></part>
</message>
<message name='postkorbKommService_sendBspNachrichtOutput'>
  <part element='tns:sendBspNachrichtOutput' name='sendBspNachrichtOutput'></part>
</message>

<message name='postkorbKommService_sendBspNachrichtNativeInput'>
  <part element='tns:sendBspNachrichtNative' name='sendBspNachrichtNative'></part>
</message>
<message name='postkorbKommService_sendBspNachrichtNativeOutput'>
  <part element='tns:sendBspNachrichtNativeOutput' name='bspQuittung'></part>
</message>

<portType name='postkorbKommPortType'>
  <operation name='sendBspNachricht' >
    <input message='tns:postkorbKommService_sendBspNachrichtInput'></input>
    <output message='tns:postkorbKommService_sendBspNachrichtOutput'></output>
  </operation>
  <operation name='sendBspNachrichtNative' >
    <input message='tns:postkorbKommService_sendBspNachrichtNativeInput'></input>
    <output message='tns:postkorbKommService_sendBspNachrichtNativeOutput'></output>
  </operation>
</portType>

<binding name="postkorbKommBinding" type="tns:postkorbKommPortType">
```

```
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />

  <operation name='sendBspNachricht'>
    <soap:operation soapAction='' />
    <input>
      <soap:body use='literal' />
    </input>
        <output>
          <soap:body use="literal"/>
        </output>
  </operation>

  <operation name='sendBspNachrichtNative'>
    <soap:operation soapAction='' />
    <input>
      <soap:body use='literal' />
    </input>
        <output>
          <soap:body use="literal"/>
        </output>
  </operation>
 </binding>

 <service name="postkorbKommService">
  <port name="postkorbKommPort" binding="tns:postkorbKommBinding">
    <soap:address location="https://webservice.bund.de/bspx-postkorb-okkomm-ws/bspservices/postkorbkomm" />
  </port>
 </service>

</definitions>
```

Die URL ist jeweils der Instanz der BundID anzupassen.

Dies entspricht folgenden Java-Signaturen:

```
byte[] sendBspNachricht(byte[] input)
```

Die Eingabe input ist ein base64-codiertes Byte-Array, welches das XML mit eingebetteter BSP-Nachricht enthält.

Hinweis: Die eingebettete Postfach-Nachricht ist damit effektiv zweifach base64-codiert.

Ausgabe ist ebenfalls ist ein base64-codiertes Byte-Array, welches ein XML mit den entsprechenden Bestätigungs- oder Fehlercode enthält.

```
String sendBspNachrichtNative(String bspNachricht)
```

Die Eingabe bspNachricht ist hier ein einfacher String, der die BSP-Nachricht enthält

Ausgabe ist ebenfalls ein einfacher String mit der BSP-Quittung.

Siehe Anhang 14.7 Annex: Webservice Schema Datei bspnachrichten-2.11.xsd

15 Anhang hinzufügen

15.1 Beispiel für eine Postfachnachricht

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Header/><SOAP-ENV:Body><ns3:sendBspNachrichtNative xmlns:ns3="urn:akdb:bsp:postkorb:komm:webservice"><bspNachricht><?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<BspNachricht xmlns="http://www.akdb.de/egov/bsp/nachrichten" version="1.2" fassung="2018-04-01">
<NachrichtenKopf>
<Identifikation.Nachricht>
<Ereignis>
<Tabelle>9001</Tabelle>
<Schluessel>BspNachricht</Schluessel>
</Ereignis>
<Erstellungszeitpunkt>2021-02-12T18:51:50.779Z</Erstellungszeitpunkt>
<NachrichtenId>1613155910779</NachrichtenId>
</Identifikation.Nachricht>
<Anwenderkennung></Anwenderkennung>
<Absender>
<PostkorbId></PostkorbId>
<Verfahren>TestDienstId</Verfahren>
<Dienst>TestDienstId</Dienst>
<Mandant>Ingolstadt</Mandant>
<Gemeindeschluessel>
<Tabelle>36</Tabelle>
<Schluessel>09161000</Schluessel>
</Gemeindeschluessel>
<Name>Paula Panther</Name>
<Anschrift>Stadtverwaltung Ingolstadt, Hauptplatz 12, 83231 Ingolstadt</Anschrift>
<Email>paula.pather@stadt-ingolstadt.de</Email>

<Telefon>08372-123123</Telefon>
<Hyperlink>http://www.akdb.de</Hyperlink>
</Absender>
<Empfaenger>
<PostkorbId>3d4f8600-751b-47d0-bf88-7a64b13287bd</PostkorbId>
<Verfahren></Verfahren>
<Dienst></Dienst>
<Mandant></Mandant>
<Gemeindeschluessel>
<Tabelle>36</Tabelle>
<Schluessel>09163000</Schluessel>
</Gemeindeschluessel>
<Name></Name>
<Anschrift></Anschrift>
</Empfaenger>
<AntwortAuf></AntwortAuf>
<WeiterleitungZu></WeiterleitungZu>
<lesebestaetigungAntwortAdresse><lesebestaetigung-antwort-adresse></lesebestaetigungAntwortAdresse>
</NachrichtenKopf>
<NachrichtenInhalt>
<Betreff>NachrichtBetreff</Betreff>
<Kategorie>
<Tabelle>9002</Tabelle>
<Schluessel>KAT_INFOBSP</Schluessel>
</Kategorie>
<StorkQaaLevel>LEVEL_1</StorkQaaLevel>
<ZuVorgang>
<VorgangsName></VorgangsName>
<VorgangsId></VorgangsId>
<VorgangStatus>
<Tabelle></Tabelle>

<Schluessel></Schluessel>
</VorgangStatus>
</ZuVorgang>
<FreiText>
<Encoding>
<Tabelle>9004</Tabelle>
<Schluessel>text/plain</Schluessel>
</Encoding>
<Text>NachrichtText</Text>
</FreiText>
<DataContainer>
<Inhalt>ZGFzaXN0ZWlua2xlaW5lRGF0YQo=</Inhalt>
<FileName>FileName</FileName>
<FileType>
<Tabelle>9005</Tabelle>
<Schluessel></Schluessel>
</FileType>
</DataContainer>
</NachrichtenInhalt>
</BspNachricht>
</bspNachricht></ns3:sendBspNachrichtNative></SOAP-ENV:Body></SOAP-ENV:Envelope>

16 Client Authentifizierung einbinden

Da der Empfangsschnittpunkt später direkt aus dem Internet erreichbar ist, ist der Endpunktpfad über den Webserver mit einer Authentifizierung über ein Client Zertifikat vorgeschaltet.

Dieses muss in der sendenden Anwendung (Online Leistung) importiert werden, damit die Anwendung den Endpunkt erfolgreich aufrufen kann.

Im Folgenden sind dazu ein Erläuterungen aufgeführt, die bei der Einrichtung der Client-Zertifikate, am Beispiel einer Java-Anwendung, unterstützten soll.

Folgendes Beispiel zeigen eine beispielhafte Implementation mittels Axis. Dies stellt keine Empfehlung oder Vorgabe seitens der AKDB dar. Die Werkzeuge und Programmiersprachen zur Umsetzung eines Senders sind komplett frei wählbar, solange sie den SOAP Standard und eine Authentifizierung per Zertifikat ermöglichen.

16.1 Hinweise zur Bildung des keystore (One-way-ssl)

Damit die jeweiligen Zertifikate innerhalb der Schlüsselspeicher adressiert werden können wird ein eindeutiger Alias vergeben. Unter dem Alias sind dann der Antragsteller-DN (DN=Distinguished Name) und public key und weitere Properties abgelegt.

```
> keytool -printcert -v -file zertifikat.cer
```

Ausgabe (_beispielhaft_):

```
CN=PortalXYZ, L=Potsdam, ST=Brandenburg, O=Brandenburgischer IT-Dienst-
leister, C=DE
```

Erläuterungen

```
CN=commonName, OU=organizationUnit, O=organizationName, L=localityName,
ST=stateName, C=country
```

Näheres zum Befehl findet sich in der Dokumentation von Oracle zu *key-tool* https://docs.oracle.com/en/java/javase/12/tools/keytool.html (Kapitel: X.500 Distinguished Names).

16.2 Hinweise zur Bildung des trustsore (Two-way-ssl)

Beim Importieren der Zertifikate setzt Keystore voraus, dass das Zertifikat vertrauenswürdig ist und die ausstellende Zertifizierungsstelle in dem Keystore bereits vorhanden ist. Das bedeutet, dass die

Zertifikate in umgekehrter Reihenfolge importiert werden, also von dem Root- bis zu dem Server-Zertifikat.

Falls die Reihenfolge nicht eingehalten wird:

Die erwähnte Bedingung des vorherigen Importierens der CA in das Keystore verursacht einen häufigen Fehler

```
keytool error: java.lang.Exception: Failed to establish chain from reply.
```

Diese Meldung bedeutet, dass in dem Keystore das ausstellende CA-Zertifikat (Intermediate) fehlt.

Für den Import gibt es weitere Möglichkeiten, siehe: https://www.sslmarket.de/ssl/verwaltung-von-zertifikaten-in-java-keystore

Als erstes wird also das Rootzertifikat importiert. Dann das Zertifikats einer möglichen Zwischenzertifizierungsstelle und schließlich das eigentliche Zertifikat bzw. der Private Schlüssel.

In dem zu definierenden Truststore müssen die Public-Keys der Identityprovider und der einreichenden Anwendungen/Portale inkl. der Rootzertifikate eingebunden werden.

```
System.setProperty("javax.net.ssl.trustStore",        System.getProperty("cata-
lina.base") + "/conf/*********truststore.jks");
 System.setProperty("javax.net.ssl.trustStorePassword", "***********");
 System.setProperty("javax.net.ssl.trustStoreType", "JKS");
```

16.3 Eigenen Proxy beachten

Damit die sendende Anwendung auch nach außen kommunizieren kann, muss, falls vorhanden, der eigene Proxyserver im Code hinterlegt werden. Dies muss dem Axis-Call explizit zugewiesen werden.

```
AxisProperties.setProperty("axis.socketSecureFactory","com.prosolu-
tion.tools.WorkExpertSSLFactory");
 System.setProperty("https.proxySet","true");
 System.setProperty("https.proxyHost","*********");
 System.setProperty("https.proxyPort","*******");
```

16.4 Komplettes Beispiel

Ein Axis-Aufruf sieht also wie folgt aus:

```
AxisProperties.setProperty("axis.socketSecureFactory","com.prosolu-
tion.tools.WorkExpertSSLFactory");
 System.setProperty("https.proxySet","true");
 System.setProperty("https.proxyHost","*********");
 System.setProperty("https.proxyPort","*******");
System.setProperty("javax.net.ssl.trustStore",      System.getProperty("cata-
lina.base") + "/conf/*********truststore.jks");
 System.setProperty("javax.net.ssl.trustStorePassword", "***********");
 System.setProperty("javax.net.ssl.trustStoreType", "JKS");
System.setProperty("javax.net.ssl.keyStore",System.getProperty("cata-
lina.base") + "/conf/" + ********keystore);
System.setProperty("javax.net.ssl.keyStorePassword", "*****************");
 System.setProperty("javax.net.ssl.keyStoreType", "JKS");
//System.setProperty("javax.net.debug", "all");
 org.apache.axis.AxisProperties.setProperty("*********", "********");
```

17 Anhänge und Verzeichnisse

17.1 Abbildungen

17.2 Verweise auf externe Dokumente

- Portalverbundvereinbarung
- Beitrittserklärung zum Portalverbund im Testbetrieb
- Beitrittserklärung zum Portalverbund im Wirkbetrieb
- Handreichung zum Nachweis der Beweiswerterhaltung bei Online Ausweis Funktion-Nutzung

17.3 Verwendete Abkürzungen

Kürzel	Ausführliche Benennung
SAML	Security Assertion Markup Language von OASIS
OASIS	Organization for the Advancement of Structured Information Standards
IDP	Identity Provider , die Authentisierungsinstanz

SP	Service-Provider, die Drittanwendung (ggf. inkl. Reverse Proxy)
SSO	Single Sign On
TLS	Transport Layer Security

17.4 Annex: Webservice Schema Dateien

17.4.1 Schemadatei - bspnachrichten-2.13.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

  Service- und Portalplattform
  AKDB München, Geschäftsfeld eGovernment

  Copyright (c) AKDB

-->
<xsd:schema targetNamespace="http://www.akdb.de/egov/bsp/nachrichten"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:bsp="http://www.akdb.de/egov/bsp/nachrichten"
 elementFormDefault="qualified">

 <xsd:element name="BspNachricht">
  <xsd:annotation>
   <xsd:appinfo>
    <title>Nachricht für die Kommunikation zwischen Bürgerservice-Portal
     und externen Fachverfahren</title>
   </xsd:appinfo>
   <xsd:documentation>Einheitliches Nachrichtenschema für die
    Kommunikation zwischen Bürgerservice-Portal und externem Verfahren.
    Nachrichten vom Bürgerservice-Portal an den Postkorb eines
    Verfahrens
    oder Nachrichten von den Verfahren an das Bürgerservice-Portal müssen
    gemäß diesem Schema aufgebaut sein.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="NachrichtenKopf" type="bsp:NachrichtenKopfType"/>
    <xsd:element name="NachrichtenInhalt" type="bsp:NachrichtenInhaltType"/>
   </xsd:sequence>
   <xsd:attribute name="version" use="required">
    <xsd:annotation>
     <xsd:documentation>Dieses Attribut kennzeichnet die
      Nachrichten-Version, z. B. "1.0", "1.1".</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleType>
```

```
      <xsd:restriction base="xsd:string">
       <xsd:enumeration value="1.1"/>
       <xsd:enumeration value="1.2"/>
       <xsd:enumeration value="1.3"/>
       <xsd:enumeration value="1.4"/>
       <xsd:enumeration value="1.5"/>
      </xsd:restriction>
    </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="fassung" use="required" >
    <xsd:annotation>
     <xsd:documentation>Dieses Attribut kennzeichnet das Datum, an dem
      die diesen Schemata im Status final produziert wurde. Format:
      YYYY-MM-DD.</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleType>
     <xsd:restriction base="xsd:string">
      <xsd:enumeration value="2017-03-15"/>
      <xsd:enumeration value="2018-04-01"/>
      <xsd:enumeration value="2018-11-01"/>
      <xsd:enumeration value="2019-06-28"/>
      <xsd:enumeration value="2020-03-15"/>
     </xsd:restriction>
    </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="produkt" type="xsd:string" use="optional">
    <xsd:annotation>
     <xsd:documentation>In diesem Attribut ist der Name des Produktes
      (der Software) einzutragen, mit dem diese Nachricht erstellt
      worden ist. z.B. BSP, PWS</xsd:documentation>
    </xsd:annotation>
   </xsd:attribute>
   <xsd:attribute name="produkthersteller" type="xsd:string" use="optional"/>
   <xsd:attribute name="produktversion" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="BspQuittung">
 <xsd:annotation>
  <xsd:appinfo>
   <title>Quittung über den Empfang einer BSO-Nachricht</title>
  </xsd:appinfo>
  <xsd:documentation>Zu einer empfangenen BSP-Nachricht wird eine
```

```
        Quittung geliefert, die bestätigt, dass die Nachricht übernommen wurde
        oder aufgrund eines technischen oder fachlichen Fehlers abgewiesen wurde.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexType>
 <xsd:sequence>
  <xsd:element name="AnnahmeErfolgreich" type="xsd:boolean" minOccurs="1" maxOccurs="1"/>
   <xsd:element name="ErgebnisStatus" type="bsp:SchluesseltabelleType" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9006 (0 (erfolgreich angenommen), 99 (sonstiger technischer Fehler), ...)</xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  <xsd:element name="ErgaenzendeHinweise" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 </xsd:sequence>
 <xsd:attribute name="version" use="required">
  <xsd:annotation>
   <xsd:documentation>Dieses Attribut kennzeichnet die
    Nachrichten-Version, z. B. "1.0", "1.1".</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1.1"/>
          <xsd:enumeration value="1.2"/>
    <xsd:enumeration value="1.3"/>
    <xsd:enumeration value="1.4"/>
    <xsd:enumeration value="1.5"/>
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:attribute>
 <xsd:attribute name="fassung" use="required">
  <xsd:annotation>
   <xsd:documentation>Dieses Attribut kennzeichnet das Datum, an dem
    die diesen Schemata im Status final produziert wurde. Format:
    YYYY-MM-DD.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="2017-03-15"/>
    <xsd:enumeration value="2018-04-01"/>
    <xsd:enumeration value="2018-11-01"/>
    <xsd:enumeration value="2019-06-28"/>
    <xsd:enumeration value="2020-03-15"/>
   </xsd:restriction>
```

```
    </xsd:simpleType>
   </xsd:attribute>
   <xsd:attribute name="produkt" type="xsd:string" use="optional">
    <xsd:annotation>
     <xsd:documentation>In diesem Attribut ist der Name des Produktes
      (der Software) einzutragen, mit dem diese Nachricht erstellt
      worden ist. z.B. BSP, PWS</xsd:documentation>
    </xsd:annotation>
   </xsd:attribute>
   <xsd:attribute name="produkthersteller" type="xsd:string" use="optional"/>
   <xsd:attribute name="produktversion" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="NachrichtenKopfType">
 <xsd:sequence>
  <xsd:element name="Identifikation.Nachricht" type="bsp:Identifikation.NachrichtType" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="Anwenderkennung" type="xsd:string" minOccurs="0">
   <xsd:annotation>
    <xsd:documentation> Die Anwenderkennung stellt Informationen über die absendende Person
     zur Verfügung und dient der Protokollierung.
     Anhand dieser Kennung kann die absendende Person identifiziert werden.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="Absender" type="bsp:AbsenderType" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="Empfaenger" type="bsp:EmpfaengerType"  maxOccurs="1" minOccurs="1"/>
  <xsd:element name="AntwortAuf" type="xsd:string" maxOccurs="1" minOccurs="0">
   <xsd:annotation>
    <xsd:documentation>Eine Nachricht kann mit einer exisiterenden
     Nachricht in Beziehung gebracht werden als Antwortnachricht oder
     weitergeleitete Nachricht. Der Bezug erfolgt hierbei über die
     NachrichtenId.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="WeiterleitungZu" type="xsd:string" minOccurs="0"/>
  <xsd:element name="lesebestaetigungAntwortAdresse" type="xsd:string" minOccurs="0"/>
 </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Identifikation.NachrichtType">
 <xsd:sequence>
  <xsd:element name="Ereignis" type="bsp:SchluesseltabelleType">
```

```
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9001, Schluessel: BspNachricht
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
   <xsd:element name="Erstellungszeitpunkt" type="xsd:dateTime" maxOccurs="1" minOccurs="1"/>
   <xsd:element name="NachrichtenId" type="xsd:string" maxOccurs="1" minOccurs="1">
    <xsd:annotation>
     <xsd:documentation>eine beliebige, eindeutige ID, die durch den
      erstellenden Client generiert wird. Zusammen mit
      Erstellungszeitpunkt und Absender kann eine Nachricht bsp-global
      eindeutig identifiziert werden.
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AbsenderType">
 <xsd:sequence>
  <xsd:element name="PostkorbId" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Verfahren" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Dienst" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Mandant" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Gemeindeschluessel" type="bsp:SchluesseltabelleType" minOccurs="0">
   <xsd:annotation>
    <xsd:documentation>Der amtliche Gemeindeschlüssel (AGS).
     Als Tabellennummer ist hier die 36 (OSCI-XMeld-Schlüsseltabelle "Amtlicher Gemeindeschluessel") zu verwenden.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="Name" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Anschrift" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Email" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Telefon" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Hyperlink" type="xsd:string" maxOccurs="1" minOccurs="0"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="EmpfaengerType">
 <xsd:sequence>
  <xsd:element name="PostkorbId" type="xsd:string" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="Verfahren" type="xsd:string" maxOccurs="1" minOccurs="0"/>
```

```
    <xsd:element name="Dienst" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="Mandant" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="Gemeindeschluessel" type="bsp:SchluesseltabelleType" minOccurs="0">
     <xsd:annotation>
      <xsd:documentation>Der amtliche Gemeindeschlüssel (AGS).
       Als Tabellennummer ist hier die 36 (OSCI-XMeld-Schlüsseltabelle "Amtlicher Gemeindeschluessel") zu verwenden.
      </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="Name" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="Anschrift" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NachrichtenInhaltType">
  <xsd:sequence>
    <xsd:element name="Betreff" type="bsp:NonEmptyString" maxOccurs="1"   minOccurs="1"/>
    <xsd:element name="Kategorie" type="bsp:SchluesseltabelleType" minOccurs="0">
     <xsd:annotation>
      <xsd:documentation>Schluesseltabelle 9002 (KAT_STATUS, KAT_INFOBSP, ...)
      </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:choice minOccurs="0">
     <xsd:element name="StorkQaaLevel" type="bsp:StorkQaaLevel"/>
     <xsd:element name="NpaGescheutzt" type="xsd:boolean" >
      <!-- deprecated, wird durch StorkQaaLevel="STORK-QAA-Level-1" ersetzt -->
      <xsd:annotation>
       <xsd:documentation>
        Diese Nachricht kann im BüsP-Postkorb nur nach
        vorheriger Anmeldung mit dem nPA gelesen werden.
       </xsd:documentation>
      </xsd:annotation>
     </xsd:element>
    </xsd:choice>
    <xsd:element name="ZuVorgang" type="bsp:ZuVorgangType" minOccurs="0"/>
    <xsd:element name="FreiText" type="bsp:FreiTextType"/>
    <xsd:element name="DataContainer" type="bsp:DataContainerType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ZuVorgangType">
  <xsd:annotation>
```

```
   <xsd:documentation>VorgangsName oder VorgangsId müssen angegeben
    werden. Es können auch beide angegeben werden.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0">
   <xsd:element name="VorgangsName" type="xsd:string"/>
   <xsd:element name="VorgangsId" type="xsd:string"/>
   <xsd:element name="VorgangStatus" type="bsp:SchluesseltabelleType">
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9003 (ST_ERHALTEN, ST_GELESEN,...)
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FreiTextType">
 <xsd:sequence>
  <xsd:element name="Encoding" type="bsp:SchluesseltabelleType" maxOccurs="1" minOccurs="1">
   <xsd:annotation>
    <xsd:documentation>Schluesseltabelle 9004 (text/plain, text/html, ...)
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="Text" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="1">
   <xsd:annotation>
    <xsd:documentation>
     Wenn als Encoding text/plain festgelegt ist, so wird die Zeichensequenz "\n" als ein Zeilenvorschub interpretiert.
     Das Backslash-Zeichen (\) wird mit einem weiteren Backslash-Zeichen entwertet.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DataContainerType">
 <xsd:sequence minOccurs="0">
  <xsd:element name="Inhalt" type="xsd:base64Binary" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="FileName" maxOccurs="1" minOccurs="0">
   <xsd:simpleType>
    <xsd:restriction base="xsd:string">
     <xsd:maxLength value="255"/>
    </xsd:restriction>
   </xsd:simpleType>
```

```
    </xsd:element>
    <xsd:element name="FileType" type="bsp:SchluesseltabelleType" maxOccurs="1" minOccurs="1">
     <xsd:annotation>
      <xsd:documentation>Schluesseltabelle 9005 (application/pdf, text/html, ...)
      </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SchluesseltabelleType">
  <xsd:annotation>
    <xsd:documentation>Dieser Datentyp wird für Schlüsselwerte benötigt.
      Mit dem Datentyp SchluesseltabelleType übermittelt man den Schlüssel
      und die Nummer der Tabelle, in der das Schlüssel-Wert Paar definiert
      worden ist.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Tabelle" type="xsd:string"/>
    <xsd:element name="Schluessel" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="StorkQaaLevel">
  <xsd:restriction base="xsd:string">
    <!-- since version 1.2 - deprecated -->
    <xsd:enumeration value="LEVEL_1"/>
    <xsd:enumeration value="LEVEL_2"/>
    <xsd:enumeration value="LEVEL_3"/>
    <xsd:enumeration value="LEVEL_4"/>
    <!--  since version 1.3 -->
    <xsd:enumeration value="STORK-QAA-Level-1"/>
    <xsd:enumeration value="STORK-QAA-Level-2"/>
    <xsd:enumeration value="STORK-QAA-Level-3"/>
    <xsd:enumeration value="STORK-QAA-Level-4"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="NonEmptyString">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:schema>
```

17.4.2 Schemadatei - bspnachrichten-schluesseltabellen-2.10.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

  Service- und Portalplattform
  AKDB München, Geschäftsfeld eGovernment

  Copyright (c) AKDB

-->
<xsd:schema targetNamespace="http://www.akdb.de/egov/bsp/nachrichten"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:bsp="http://www.akdb.de/egov/bsp/nachrichten"
 elementFormDefault="qualified">

 <xsd:element name="BspNachricht">
  <xsd:annotation>
   <xsd:appinfo>
    <title>Nachricht für die Kommunikation zwischen Bürgerservice-Portal
     und externen Fachverfahren</title>
   </xsd:appinfo>
   <xsd:documentation>Einheitliches Nachrichtenschema für die
    Kommunikation zwischen Bürgerservice-Portal und externem Verfahren.
    Nachrichten vom Bürgerservice-Portal an den Postkorb eines
    Verfahrens
    oder Nachrichten von den Verfahren an das Bürgerservice-Portal müssen
    gemäß diesem Schema aufgebaut sein.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="NachrichtenKopf" type="bsp:NachrichtenKopfType"/>
    <xsd:element name="NachrichtenInhalt" type="bsp:NachrichtenInhaltType"/>
   </xsd:sequence>
   <xsd:attribute name="version" use="required">
    <xsd:annotation>
     <xsd:documentation>Dieses Attribut kennzeichnet die
      Nachrichten-Version, z. B. "1.0", "1.1".</xsd:documentation>
```

```
    </xsd:annotation>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="1.1"/>
        <xsd:enumeration value="1.2"/>
        <xsd:enumeration value="1.3"/>
        <xsd:enumeration value="1.4"/>
        <xsd:enumeration value="1.5"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="fassung" use="required" >
    <xsd:annotation>
      <xsd:documentation>Dieses Attribut kennzeichnet das Datum, an dem
        die diesen Schemata im Status final produziert wurde. Format:
        YYYY-MM-DD.</xsd:documentation>
    </xsd:annotation>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="2017-03-15"/>
        <xsd:enumeration value="2018-04-01"/>
        <xsd:enumeration value="2018-11-01"/>
        <xsd:enumeration value="2019-06-28"/>
        <xsd:enumeration value="2020-03-15"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="produkt" type="xsd:string" use="optional">
    <xsd:annotation>
      <xsd:documentation>In diesem Attribut ist der Name des Produktes
        (der Software) einzutragen, mit dem diese Nachricht erstellt
        worden ist. z.B. BSP, PWS</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="produkthersteller" type="xsd:string" use="optional"/>
  <xsd:attribute name="produktversion" type="xsd:string" use="optional"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="BspQuittung">
 <xsd:annotation>
  <xsd:appinfo>
   <title>Quittung über den Empfang einer BSO-Nachricht</title>
```

```
 </xsd:appinfo>
 <xsd:documentation>Zu einer empfangenen BSP-Nachricht wird eine
        Quittung geliefert, die bestätigt, dass die Nachricht übernommen wurde
        oder aufgrund eines technischen oder fachlichen Fehlers abgewiesen wurde.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexType>
 <xsd:sequence>
  <xsd:element name="AnnahmeErfolgreich" type="xsd:boolean" minOccurs="1" maxOccurs="1"/>
   <xsd:element name="ErgebnisStatus" type="bsp:SchluesseltabelleType" minOccurs="1" maxOccurs="1">
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9006 (0 (erfolgreich angenommen), 99 (sonstiger technischer Fehler), ...)</xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  <xsd:element name="ErgaenzendeHinweise" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 </xsd:sequence>
 <xsd:attribute name="version" use="required">
  <xsd:annotation>
   <xsd:documentation>Dieses Attribut kennzeichnet die
    Nachrichten-Version, z. B. "1.0", "1.1".</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="1.1"/>
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:attribute>
 <xsd:attribute name="fassung" use="required">
  <xsd:annotation>
   <xsd:documentation>Dieses Attribut kennzeichnet das Datum, an dem
    die diesen Schemata im Status final produziert wurde. Format:
    YYYY-MM-DD.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
   <xsd:restriction base="xsd:string">
    <xsd:enumeration value="2017-03-15"/>
   </xsd:restriction>
  </xsd:simpleType>
 </xsd:attribute>
 <xsd:attribute name="produkt" type="xsd:string" use="optional">
  <xsd:annotation>
   <xsd:documentation>In diesem Attribut ist der Name des Produktes
    (der Software) einzutragen, mit dem diese Nachricht erstellt
```

```
        worden ist. z.B. BSP, PWS</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="produkthersteller" type="xsd:string" use="optional"/>
    <xsd:attribute name="produktversion" type="xsd:string" use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="NachrichtenKopfType">
  <xsd:sequence>
    <xsd:element name="Identifikation.Nachricht" type="bsp:Identifikation.NachrichtType" maxOccurs="1" minOccurs="1"/>
    <xsd:element name="Anwenderkennung" type="xsd:string" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation> Die Anwenderkennung stellt Informationen über die absendende Person
        zur Verfügung und dient der Protokollierung.
        Anhand dieser Kennung kann die absendende Person identifiziert werden.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Absender" type="bsp:AbsenderType" maxOccurs="1" minOccurs="1"/>
    <xsd:element name="Empfaenger" type="bsp:EmpfaengerType"  maxOccurs="1" minOccurs="1"/>
    <xsd:element name="AntwortAuf" type="xsd:string" maxOccurs="1" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Eine Nachricht kann mit einer exisiterenden
          Nachricht in Beziehung gebracht werden als Antwortnachricht oder
          weitergeleitete Nachricht. Der Bezug erfolgt hierbei über die
          NachrichtenId.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="WeiterleitungZu" type="xsd:string" minOccurs="0"/>
    <xsd:element name="lesebestaetigungAntwortAdresse" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Identifikation.NachrichtType">
  <xsd:sequence>
    <xsd:element name="Ereignis" type="bsp:SchluesseltabelleType">
      <xsd:annotation>
        <xsd:documentation>Schluesseltabelle 9001, Schluessel: BspNachricht
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Erstellungszeitpunkt" type="xsd:dateTime" maxOccurs="1" minOccurs="1"/>
```

```
   <xsd:element name="NachrichtenId" type="xsd:string" maxOccurs="1" minOccurs="1">
    <xsd:annotation>
     <xsd:documentation>eine beliebige, eindeutige ID, die durch den
      erstellenden Client generiert wird. Zusammen mit
      Erstellungszeitpunkt und Absender kann eine Nachricht bsp-global
      eindeutig identifiziert werden.
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AbsenderType">
 <xsd:sequence>
  <xsd:element name="PostkorbId" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Verfahren" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Dienst" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Mandant" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Gemeindeschluessel" type="bsp:SchluesseltabelleType" minOccurs="0">
   <xsd:annotation>
    <xsd:documentation>Der amtliche Gemeindeschlüssel (AGS).
     Als Tabellennummer ist hier die 36 (OSCI-XMeld-Schlüsseltabelle "Amtlicher Gemeindeschluessel") zu verwenden.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="Name" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Anschrift" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Email" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Telefon" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Hyperlink" type="xsd:string" maxOccurs="1" minOccurs="0"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="EmpfaengerType">
 <xsd:sequence>
  <xsd:element name="PostkorbId" type="xsd:string" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="Verfahren" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Dienst" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Mandant" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  <xsd:element name="Gemeindeschluessel" type="bsp:SchluesseltabelleType" minOccurs="0">
   <xsd:annotation>
    <xsd:documentation>Der amtliche Gemeindeschlüssel (AGS).
     Als Tabellennummer ist hier die 36 (OSCI-XMeld-Schlüsseltabelle "Amtlicher Gemeindeschluessel") zu verwenden.
```

```
      </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
   <xsd:element name="Name" type="xsd:string" maxOccurs="1" minOccurs="0"/>
   <xsd:element name="Anschrift" type="xsd:string" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NachrichtenInhaltType">
  <xsd:sequence>
   <xsd:element name="Betreff" type="bsp:NonEmptyString" maxOccurs="1"   minOccurs="1"/>
   <xsd:element name="Kategorie" type="bsp:SchluesseltabelleType" minOccurs="0">
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9002 (KAT_STATUS, KAT_INFOBSP, ...)
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
   <xsd:choice minOccurs="0">
    <xsd:element name="StorkQaaLevel" type="bsp:StorkQaaLevel"/>
    <xsd:element name="NpaGescheutzt" type="xsd:boolean" >
     <!-- deprecated, wird durch StorkQaaLevel="STORK-QAA-Level-1" ersetzt -->
     <xsd:annotation>
      <xsd:documentation>
       Diese Nachricht kann im BüsP-Postkorb nur nach
       vorheriger Anmeldung mit dem nPA gelesen werden.
      </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
   </xsd:choice>
   <xsd:element name="ZuVorgang" type="bsp:ZuVorgangType" minOccurs="0"/>
   <xsd:element name="FreiText" type="bsp:FreiTextType"/>
   <xsd:element name="DataContainer" type="bsp:DataContainerType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ZuVorgangType">
  <xsd:annotation>
   <xsd:documentation>VorgangsName oder VorgangsId müssen angegeben
    werden. Es können auch beide angegeben werden.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence minOccurs="0">
   <xsd:element name="VorgangsName" type="xsd:string"/>
   <xsd:element name="VorgangsId" type="xsd:string"/>
```

```
   <xsd:element name="VorgangStatus" type="bsp:SchluesseltabelleType">
    <xsd:annotation>
     <xsd:documentation>Schluesseltabelle 9003 (ST_ERHALTEN, ST_GELESEN,...)
     </xsd:documentation>
    </xsd:annotation>
   </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FreiTextType">
 <xsd:sequence>
  <xsd:element name="Encoding" type="bsp:SchluesseltabelleType" maxOccurs="1" minOccurs="1">
   <xsd:annotation>
    <xsd:documentation>Schluesseltabelle 9004 (text/plain, text/html, ...)
     </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
  <xsd:element name="Text" type="bsp:NonEmptyString" maxOccurs="1" minOccurs="1">
   <xsd:annotation>
    <xsd:documentation>
     Wenn als Encoding text/plain festgelegt ist, so wird die Zeichensequenz "\n" als ein Zeilenvorschub interpretiert.
     Das Backslash-Zeichen (\) wird mit einem weiteren Backslash-Zeichen entwertet.
    </xsd:documentation>
   </xsd:annotation>
  </xsd:element>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DataContainerType">
 <xsd:sequence minOccurs="0">
  <xsd:element name="Inhalt" type="xsd:base64Binary" maxOccurs="1" minOccurs="1"/>
  <xsd:element name="FileName" maxOccurs="1" minOccurs="0">
    <xsd:simpleType>
     <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
     </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="FileType" type="bsp:SchluesseltabelleType" maxOccurs="1" minOccurs="1">
   <xsd:annotation>
    <xsd:documentation>Schluesseltabelle 9005 (application/pdf, text/html, ...)
     </xsd:documentation>
   </xsd:annotation>
```

```
    </xsd:element>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="SchluesseltabelleType">
  <xsd:annotation>
   <xsd:documentation>Dieser Datentyp wird für Schlüsselwerte benötigt.
    Mit dem Datentyp SchluesseltabelleType übermittelt man den Schlüssel
    und die Nummer der Tabelle, in der das Schlüssel-Wert Paar definiert
    worden ist.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
   <xsd:element name="Tabelle" type="xsd:string"/>
   <xsd:element name="Schluessel" type="xsd:string"/>
  </xsd:sequence>
 </xsd:complexType>

 <xsd:simpleType name="StorkQaaLevel">
  <xsd:restriction base="xsd:string">
   <!-- since version 1.2 - deprecated -->
   <xsd:enumeration value="LEVEL_1"/>
   <xsd:enumeration value="LEVEL_2"/>
   <xsd:enumeration value="LEVEL_3"/>
   <xsd:enumeration value="LEVEL_4"/>
   <!--  since version 1.3 -->
   <xsd:enumeration value="STORK-QAA-Level-1"/>
   <xsd:enumeration value="STORK-QAA-Level-2"/>
   <xsd:enumeration value="STORK-QAA-Level-3"/>
   <xsd:enumeration value="STORK-QAA-Level-4"/>
  </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="NonEmptyString">
  <xsd:restriction base="xsd:string">
   <xsd:minLength value="1"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:schema>
```

18 Quellen

[1] OASIS SAML: http://saml.xml.org/saml-specifications

[2] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindeststandards/Mindeststandard_BSI_TLS_1_ 2_Version_1_0.pdf?__blob=publicationFile

[3] http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[4] http://www.capcourse.com/Library/OpenSAML/index.html

[5] http://forgerock.com/products/open-identity-stack/openig/

[6] https://shibboleth.atlassian.net/wiki/spaces/SP3/overview

[7] Beispiel am Ende der Herstellerdokumentation: https://wiki.shibboleth.net/confluence/display/SP3/SAML2+SessionInitiator

[8] https://tools.ietf.org/html/rfc4519

[9] https://tools.ietf.org/html/rfc4524

[10] https://tools.ietf.org/html/draft-gryphon-ldap-schema-vcard4-00#section-3.5

[11] https://ec.europa.eu/digital-agenda/en/trust-services-and-eid

[12] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG

[13] https://www.eid-stork2.eu/index.php?option=com_phocadownload&view=file&id=6:d32-qaa-status-report&Itemid=175) [Abrufdatum 14.06.2018]

[14] https://calver.org

[15[https://tools.ietf.org/html/rfc4519#section-2.25

[16] http://docs.oracle.com/cd/E19462-01/819-4670/gbanp/index.html

ozg-umsetzung.de