Pflichtenheft:

Microservice zum Daten-Transfer von App-Betreibern an Kundenvertragspartner

Im Projektkontext auch als „Meldesystem“ oder „Middleware“ bezeichnet.

Version 2.2

Stand: 20.10.2026

Autoren:

Konstantin Schaible (SSB AG)

René Piesker (VVS GmbH)

Daniel Holz (bodo GmbH)

Werner Mies (DING GmbH)

Harald Bender (NVBW GmbH)

Tim Weiskopf (NVBW GmbH)

Sebastian Neifer (NVBW GmbH)

Andreas Schmid (Almato AG)

Dr. Tobias Henninger (NVBW GmbH)

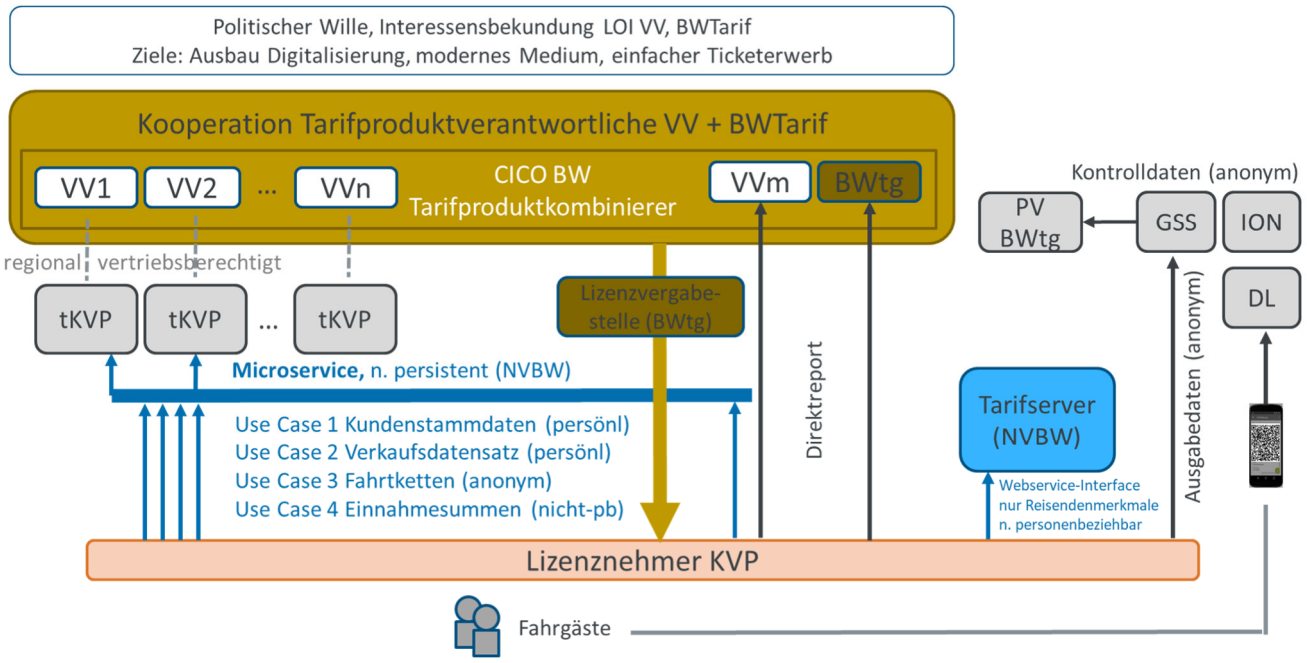
Historie

Version	Datum	Verantwortlicher	Anmerkungen
1.0	29.09.2021	S. Neifer	
1.1	31.07.2022	S. Neifer	
1.2	25.10.2022	A. Schmid	Korrektur der JSON-Beispieldaten
1.3	06.12.2022	T. Henninger	Übersichtsbild erneuert und Einführung zu den Uses Cases verfasst
1.4	07.12.2022	A. Schmid	Datenformat vereinheitlicht, Beispiele angepasst, Beschreibung Bool Attribute angepasst.
1.5	09.12.2022	T. Henninger	Testumgebung eingesetzt
1.6	07.02.2023	T. Henninger	Opt-In-Vorgaben eingesetzt
1.7	15.05.2023	T. Henninger	Beschreibungsfehler korrigiert
1.8	17.07.2023	A. Schmid	Anpassungen der Optional-Parameter im Customer-Objekt, Erweiterungen Test-Flag, Kommunikation Zielserver
1.9	04.08.2023	A. Schmid	Anpassung der Fahrketten Beispiel JSON
1.91	08.08.2023	T. Henninger	Änderungen angenommen, Ergänzung von Beschreibungen bei Technische Datenübermittlung
1.92	09.08.2023	T. Henninger	Tarifserver-Uhrzeiten erläutert
1.93	16.09.2023	A. Schmid	Umbenennung Halteliste in Haltestellenliste und Anpassung Beispieldaten Use-Case 3
1.94	17.08.2023	A. Schmid	Line ist nun Pflicht, Verkehrsmittel sind jetzt auf derselben Ebene wie die Liniendaten, JSON aktualisiert.
1.95	04.09.2023	T. Henninger	TicketTypDesc auf Länge 100, Aenderungszeitpunkt und Verkaufszeitpunkt im Format „dd.mm.yyyy hh:mm:ss“, da nicht vom Tarifserver, sondern von den Lizenznehmersystemen gesetzt
1.96	05.09.2023	A. Schmid	Korrekturen der Feldlänge der Attribute Äenderungszeitpunkt und Verkaufszeitpunkt auf 19. Anpassung der Beispieldaten.
1.97	05.03.2024	A. Schmid	Ergänzungen Post Limit und Service URLs
1.98	21.02.2024	A. Schmid	Länge der AbrechnungsPositionID und AbrechnungsID angepasst
1.99	23.05.2024	A. Schmid	Ergänzung Produktivsystem und IP-Adressprüfung
2.0	27.05.2024	A. Schmid	Ergänzung Outgoing IP-Adressen
2.1	27.03.2026	T. Henninger	Erweiterung Use Case 3 für bwtarif
2.2	20.10.2026	T. Henninger	Konfiguration der Dateinamen mit KVP_OrgID usw.

Inhalt

Historie 2
Einführung und Übersicht CICO BW 5
Use Cases 6
Übersicht 6
Datenschutz und Opt-Ins in der Endnutzung 7
Use Case 1: Kundenstammdaten 8
Häufigkeit der Übertragung 8
Daten 8
Use Case 2: Verkaufsdatensätze (Marketing/Vertrieb + Abrechnung/Einnahme) 10
Häufigkeit der Übertragung mindestens einmal im Monat kumuliert 11
Daten 11
Use Case 3: Fahrtketten-Datensätze: 14
Häufigkeit der Übertragung mindestens einmal im Monat kumuliert 15
Daten 15
Use Case 4: Einnahmesumme-Datensätze: 19
Häufigkeit der Übertragung mindestens einmal im Monat kumuliert 19
Daten 19
Technische Datenübermittlung 23
a) Vom Lizenznehmer (App-Betreiber) an Microservice (NVBW) 23
Endpoint Testsystem 23
Endpoint Produktivsystem 23
Übertragungs-Limit 24
b) Konfiguration des Microservice 24
Datenstruktur 24
Häufigkeit der Datenübertragung 25
c) Anforderung an Datenabnehmer/Ziel 25
d) Fehlerhandling 26

Das Übersichtsbild zeigt die funktionale Gesamtarchitektur von CICO BW und die Verortung des hier beschriebenen Microservices.



Die Politik hat sich zum Ziel gesetzt, die Digitalisierung speziell auch im Mobilitätssektor voranzubringen und den Modal Split-Anteil des ÖPNV bis 2030 deutlich zu erhöhen (ÖPNV-Strategie 2030 des Ministeriums für Verkehr Baden-Württemberg). Dazu verhelfen einfache Kundenzugänge in Form moderner Medien wie Smartphones und ein Abbau von Zugangshemmnissen bei Ticketerwerb und in der Tariflandschaft.

Vor diesem Hintergrund hat sich eine Kooperation der Verkehrsverbünde (VV) und der Baden-Württemberg-Tarif GmbH (BWtg) gebildet, um im Land ein Check-In-Check-Out-Produkt auf den Markt bringen (CICO BW). Fahrgäste sollen bei Fahrtantritt auf ihrem Smartphone einchecken und am Fahrtende wieder auschecken, ohne sich Gedanken über die einzelnen Tarife machen zu müssen. Bei CICO BW handelt es sich nicht eigentlich um ein neues Tarifprodukt, sondern um einen Tarifproduktkombinierer. Dieser kombiniert einzelne Tarifprodukte der Verbünde und des BW Tarifs so, als hätten sich die Fahrgäste die Einzelprodukte in einer optimalen Weise selbst zusammengekauft. Die Logik dazu ist in einem Tarifserver untergebracht, der von Fraunhofer IVI geliefert und bei der NVBW gehostet wird. Verbünde und die BWtg sind dabei Produktverantwortliche (PV) im Sinne der VDV-KA – und zwar für ihre je eigenen zugelieferten Teilkomponenten. Die CICO-Kooperation hat entschieden, CICO BW nicht selbst zu vertreiben, sondern die Vertriebsrechte dazu an beliebig viele Lizenznehmer zu vergeben. Zur Unterstützung des Lizenzierungsprocedere hat die BW Tarif GmbH die Rolle einer Lizenzvergabestelle eingenommen. Als Fahrtberechtigung wird ein VDV-konformer 2D-Barcode auf dem Smartphone der Fahrgäste ausgegeben. Eine Kontrolle erfolgt mit der Infrastruktur, wie sie sich bereits für die elektronische Kontrolle des BWtarifs im Aufbau befindet (rechter AST im Schaubild mit der VDV-KA-Rolle der kontrollierenden Dienstleister (DL), in der Regel die Verkehrsunternehmen. Sie speisen ihre Kontrolldaten über das Interoperabilitätsnetzwerk (ION) der VDV-KA bzw. die regionale Vermittlungsstelle (GSS) in die PV-Systeme der Produktverantwortlichen, sofern diese vorhanden sind.

Die Eigenschaft von CICO BW als ein Kombinierer von einzelnen Tarifprodukten bringt es nun mit sich, dass die vertreibenden Lizenznehmer ihre Einnahmen mit genau jedem einzelnen Tarifproduktgeber abrechnen müssen, dessen Produkte sie verkauft haben. Liegt die Reisekette innerhalb eines Verbundes, kommen die jeweiligen Verbundtarife zur Anwendung (z. B. VVS), ist die Reisekette verbundübergreifend, so gilt der BW Tarif. Lizenzgeber müssen somit gegenüber mehr als 20 einzelnen Tarifproduktgebern (Produktverantwortlichen PV) abrechnen und die entsprechenden Daten übermitteln. BW Tarif GmbH setzt dabei auf die direkte bilaterale Datenübermittlung, wie sie bereits aus dem Verkauf von bwtarif besteht. Auch einzelne Verbünde setzen auf eine bilaterale Abrechnung, vor allem, wenn sie selbst für den Vertrieb verantwortlich sind. In der Regel jedoch übernehmen in einem Verbund andere Organisationen den Vertrieb: Die für sie handelnden territorialen Kundenvertragspartner (tKVP).

Damit CICO-Lizenznehmer nun nicht die Daten sämtlicher tKVP an jeden einzelnen tKVP separat verschicken müssen, wird dieser Microservice eingerichtet. Er übernimmt die Aufgabe des Versands der Daten an die richtigen Empfänger. Als Unterscheidungsmerkmal, an welchen tKVP die Daten versandt werden sollen, bedient er sich der ORG_ID des PV, also des dahinterstehenden tarifgebenden Verkehrsverbunds. Die ORD_ID ist ein Datum des eTicketing-Standards der VDV-Kernapplikation.

Use Cases

Übersicht

Die Lizenznehmer haben an die tKVP Daten zur Abrechnung, zum Support und zu Marketingzwecken zu übermitteln. Dazu wurden im Einzelnen die folgenden vier Use Cases definiert:

1. Use Case 1: Kundenstammdaten (persönlich)
2. Use Case 2: Verkaufsdatensätze (personenbeziehbar)
3. Use Case 3: Fahrtketten (anonym)
4. Use Case 4: Einnahmesummen (nicht personenbeziehbar)

Zur Motivation der einzelnen Use Cases kann folgendes ausgesagt werden: Zur monatlichen Abrechnung in jedem Falle benötigt werden die Monatssummen in Use Case 4. Die tkVP werden diese in die verbundinterne Einnahmeaufteilung einspeisen müssen. Da viele tKVP die Einnahmen jedoch aus Einzelverkäufen aufsummieren und nicht zuletzt zu Marketingzwecken Kenntnis ihrer Kunden haben möchten, wird Use Case 2 benötigt, der den Verkaufsdatensatz mit den Einzelverkäufen enthält. Die zugehörenden und nicht häufig wechselnden Kundenstammdaten sind in Use Case 1 gefasst. Die anonymen Fahrtketten von Use Case 3 werden zu Zwecken der Verkehrs- und Produktplanung erfasst.

Wichtig: Es darf aus Datenschutzgründen keine Speicherung von Daten im Microservice der NVBW erfolgen. Die Datenverarbeitung muss nicht-persistent vonstattengehen.

Im Folgenden soll nun der Inhalt und die Struktur der einzelnen Use Case beschrieben werden. Werden persönliche Daten via Microservice übermittelt, so ist stets auch die Kundeneinwilligung zu übermitteln. Genauere Betrachtungen zu den Opt-Ins bzw- Opt-Outs werden daher im Folgenden noch vorangestellt.

Datenschutz und Opt-Ins in der Endnutzung

Es besteht bei den lizenzgebenden Verbünden (Produktverantwortliche PV) der Wunsch, Kundendaten aus Fahrscheinverkäufen für Marketing und individualisierte Werbekampagnen einzusetzen. Sie begründen dies datenschutzrechtlich mit dem Zweck eines „berechtigten Interesses“. (Lediglich die ebenfalls als PV fungierende BWTG signalisierte kein Interesse an einer diesbezüglichen Verwendung von Kundendaten.) Der Landesbeauftragten für Datenschutz und Informationsfreiheit (LfDI) konnte der Argumentation eines „berechtigten Interesses“ jedoch nicht folgen und hat eine Empfehlung ausgesprochen, die Weitergabe von Kundendaten von den Lizenznehmern zu den tKVP an Opt-Ins der Fahrgäste zu knüpfen. Die Lizenzgeber haben sich dieser Empfehlung angeschlossen.

Der Microservice ist daher von den Lizenznehmern zur Weitergabe so einzusetzen, dass persönliche Daten nur bei vorliegenden Opt-Ins an die tKVP weitergegeben werden. Für die Aufnahme der verschiedenen Kundeneinwilligungen wurden in Use Case 1 die üblichen Felder in die Übergabestruktur aufgenommen. Dies sind:

- Einwilligung zur telefonischen Kontaktaufnahme (Opt-In)
- Einwilligung zur Kontaktaufnahme per E-Mail (Opt-In)
- Einwilligung der Datennutzung zur Marktforschung (Opt-In)
- Verweigerung postalischer Werbung (Opt-Out)

Der Use Case 1 Kundenstammdaten muss daher nur übermittelt werden, wenn mind. eine der oben genannten Einwilligungen oder keine Verweigerung der postalischen Werbung vorliegt. Zusätzlich darf selbstverständlich kein kundenseitiger Wunsch zur Datenlöschung vorliegen. Wird Use Case 1 übermittelt, gelten die folgenden Einschränkungen:

- Übermittlung der Telefonnummer nur bei entsprechendem Opt-In
- Übermittlung der E-Mail-Adresse nur bei entsprechendem Opt-In
- Übermittlung der Wohnadresse nur ohne Verweigerung postalischer Werbung

Use Case 2 übermittelt den Verkaufsdatensatz granular je Verkaufsvorgang. Bis auf die KundenID bei den Verkäufen enthält Use Case 2 keine persönlichen Daten. Der Verkaufsdatensatz ist allerdings personenbeziehbar, sofern die KundenID übermittelt wird. Er wird damit zu seinem persönlichen Verkaufsdatensatz. Ein persönlicher Verkaufsdatensatz wird jedoch nur zum Zwecke von Marketing und Vertrieb übermittelt. Daher ist ein Opt-In bei Marktforschung die zwingende Voraussetzung, dass bei Use Case 2 die KundenID übermittelt werden darf.

Doch selbst wenn die KundenID in Use Case 2 nicht übermittelt wird, muss Use Case 2 ohne KundenID übermittelt werden, denn ein tKVP benötigt diese Daten für die Abrechnung gegenüber dem PV, in der Regel zur Einspeisung in die Einnahmeaufteilung (Ausnahme: ein tKVP begnügt sich mit den Einnahmesumme in Use Case 4). Daher muss Use Case 2 alle abrechnungsrelevanten Daten eines Verkaufsdatensatzes enthalten (Verkaufszeitpunkt, Relation, Eindeutigkeit des Fahrscheins). Diese Information ist auch in den anoymen Daten von Use Case 2 ohne KundenID enthalten.

Use Case 1: Kundenstammdaten

Übermittlung der Kundenstammdaten vom Lizenznehmer zum territorial verantwortlichen Kundenvertragspartner.

Beispiel:

- Lizenznehmer betreibt App,
- Kunde registriert sich in App, fährt im Verbundgebiet der VVS GmbH,
- Kundenstammdaten werden über den Microservice der NVBW an die SSB AG übertragen.

Häufigkeit der Übertragung

mindestens 1x monatlich

Daten

Name	Key	Type	Länge	Value	Beschreibung
Header	Pflicht	String	30	Kunde	Legt fest, welcher Nachrichtenart gesendet wird
Testdatensatz	Optional	Boolean		true, false	Möglichkeit, die Übermittlung und Verarbeitung zu testen. Der Datensatz wird dann in der Abrechnung am Ziel nicht berücksichtigt.
ApiVersion	ReadOnly	String	30	X.Y	Die Versionsnummer wird vom Microservice in den Datensatz geschrieben. Bereits existierende Werte werden überschrieben.
KvpOrgID	Pflicht	Integer	5		Org-ID nach VDV-KA des Lizenznehmers
PvOrgID	Pflicht	Integer	5		Org-ID des Verkehrsverbundes in dem das Ticket verkauft wird
AbrechnungsPeriode	Pflicht	String	7	08-2021	Monat, für welchen die Tickets abgerechnet wurden Format: mm-yyyy z.B. 08-2021
Kundenliste					JSON-Struktur der Kunden

Name	Key	Type	Länge	Value	Beschreibung

KundenID	Pflicht	String	50		Eindeutige sich nicht verändernde Kundennummer des
					Benutzers beim Lizenznehmer
Titel	Optional	String	30		Titel der Person
Name	Pflicht	String	50		Nachname der Person
Vorname	Pflicht	String	50		Vorname der Person
Strasse	Optional	String	50		Adresse Straßenname
Hausnummer	Optional	String	30		Adresse Hausnummer
Postleitzahl	Optional	String	30		Adresse Postleitzahl
Ort	Optional	String	30		Adresse Wohnort
Land	Optional	String	30		Länderklartext wenn abweichend von Deutschland. Bei Deutschland nicht notwendig.
TelNr	Optional	String	50		Telefonische Erreichbarkeit der Person
Mailadresse	Optional	String	50		Erreichbarkeit per E-Mail der Person
Geburtsdatum	Pflicht	String	10	31.07.1999	Geburtsdatum im Format dd.mm.yyyy
EinwilligungWerbungTel	Pflicht	Boolean	1		Einwilligung zur telefonischen Kontaktaufnahme TRUE oder FALSE
EinwilligungWerbungEmail	Pflicht	Boolean	1		Einwilligung zur Kontaktaufnahme per E-Mail
EinwilligungMarktforschung	Pflicht	Boolean	1		Einwilligung zu Marktforschungszwecken
EntzugWerbungPost	Pflicht	Boolean	1		Verweigerung postalischer Werbung
Aenderungszeitpunkt	Pflicht	String	19	02.08.2023 07:03:22	Datum und Uhrzeit, zu welcher der Benutzer zuletzt geändert wurde Format: „dd.mm.yyyy hh:mm:ss“ Angabe vom Lizenznehmer
Loeschkennzeichen	Pflicht	Boolean	1		Information ob der Kundendatensatz gelöscht werden soll TRUE oder FALSE

Beispiel:

```
curl --location 'https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/customer' \
--header 'x-api-key: WZaXTzjOju3OCIH3ZqL0DaRUrZ6Qd9F3a2UsStUw' \
--header 'Content-Type: application/json' \
--data-raw '{
  "Header": "Kunde",
  "KvpOrgID": 6233,
  "PvOrgID": 6234,
  "Abrechnungsperiode": "08-2021",
  "Kundenliste": [
    {
      "KundenID": "Test123",
      "Titel": "Dr.",
      "Name": "Mustermann",
      "Vorname": "Max",
      "Strasse": "Musterstraße",
      "Hausnummer": "37/F",
      "Postleitzahl": "54321",
      "Ort": "Musterstadt",
      "Land": "Schweiz",
      "TelNr": "+41 5461  6441258",
      "Mailadresse": "test@mail.ch",
      "Geburtsdatum": "01.03.2001",
      "EinwilligungWerbungTel": true,
      "EinwilligungWerbungEmail": false,
      "EinwilligungMarktforschung": false,
      "EntzugWerbungPost": false,
      "Aenderungszeitpunkt": "27.08.2021 13:24:45",
      "Loeschkennzeichen": false
    },
    {
      "KundenID": "Test124",
      "Titel": "Dr. Prof. ",
      "Name": "Musterfrau",
      "Vorname": "Ruth",
      "Strasse": "Musterstraße",
      "Hausnummer": "31/F",
      "Postleitzahl": "34567",
      "Ort": "Musterstadt",
      "TelNr": "+41 5461  6441258",
      "Mailadresse": "test@mail.de",
      "Geburtsdatum": "01.03.2001",
      "EinwilligungWerbungTel": false,
      "EinwilligungWerbungEmail": false,
      "EinwilligungMarktforschung": false,
      "EntzugWerbungPost": true,
      "Aenderungszeitpunkt": "27.08.2021 13:24:45",
      "Loeschkennzeichen": false
    }
  ]
}
'
```

Use Case 2: Verkaufsdatensätze (Marketing/Vertrieb + Abrechnung/Einnahme)

Übermittlung der einzelnen Verkaufstransaktionen vom Lizenznehmer zum territorial verantwortlichen Kundenvertragspartner, für die Ermittlung der Werte zur Meldung an die Einnahmeaufteilung (Einnahmemeldung) und zu Zwecken des Marketing.

Beispiel:

- Lizenznehmer betreibt App,
- Kunde fährt mit App im Verbundgebiet der VVS GmbH,

- Gefahrene Fahrtkettendaten führen zu einem VVS Tarifprodukt
- Ein Ticket der VVS GmbH wird dem Kunden abgerechnet
- Abrechnungsdaten werden über die NVBW an die SSB AG übertragen. Verkaufte Tarifprodukte werden über die NVBW an die SSB AG übertragen.
- Die SSB nutzt die Übermittlung der verkauften Tarifprodukte für Einnahmemeldungen
- Die SSB kann die Verkaufstransaktionsdaten für Marketing und Vertrieb nutzen

Häufigkeit der Übertragung

mindestens einmal im Monat kumuliert

Daten

Kopf

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Header	Pflicht	String	30	Verkauf	Legt fest, welcher Nachrichtenart gesendet wird
Testdatensatz	Optional	Boolean		true, false	Möglichkeit, die Übermittlung und Verarbeitung zu testen. Der Datensatz wird dann in der Abrechnung am Ziel nicht berücksichtigt.
ApiVersion	ReadOnly	String	30	X.Y	Die Versionsnummer wird vom Microservice in den Datensatz geschrieben. Bereits existierende Werte werden überschrieben.
KvpOrgID	Pflicht	Integer	5		Org-ID nach VDV-KA des Lizenznehmers
PvOrgID	Pflicht	Integer	5		Org-ID des Verkehrsverbundes
AbrechnungsID	Pflicht	String	50		Abrechnungskopfidentifikation (eindeutige ID der Abrechnung in der Abrechnungsperiode)
AbrechnungsPeriode	Pflicht	String	7	08-2021	Monat für welchen die Tickets abgerechnet wurden Format: mm-yyyy z.B. 08-2021 Info: Zeitraum wird vom Lizenznehmer festgelegt
Verkaufsliste	Pflicht				JSON-Struktur mit Verkäufen

Verkaeufe

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
AbrechnungsPositionID	Pflicht	Integer	16		Identifier des Datensatzes
KundenID	Optional	String	50		Kundennummer des Benutzers beim

					Lizenznehmer (bei Werbesperre Marktforschung nicht übermitteln)
TicketTypID	Pflicht	String	20		ID des Tickettyps, z.B. 101 oder Fehlercode für unbekannte Restposten (z.B. aus Tarifserverfehlerdiffer enzen)

TicketTypDesc	Pflicht	String	100		Bezeichnung des Tickettyps, z.B.: Einzelticket Erwachsener oder Erstattung
PreisStufe	Optio nal	String	20	3	Preisstufe des Produktes
PreisEinzel	Pflicht	Integer	12	540	Preis für ein Ticket in Cent ohne Trennzeichen
StartAbrechnung	Optio nal	String	50	100 (Ulm)	Start des abgerechneten Tarifprodukts
ZielAbrechnung	Optio nal	String	50	300 (Erbach)	Ziel des abgerechneten Tarifprodukts
RaeumlicheGueltigkeit	Optio nal	String	75		Räumliche Gültigkeit des Tarifproduktes
BeginnGueltigkeit	Pflicht	String	25	2023-06-23T23:12:00+02:00	Gültigkeitsbeginn des Tickets (Date-Time) oder Erstattungszeitpunkt in local time Format: UTC/GMT+1 (MEZ) im Winter UTC/GMT+2 (MESZ) im Sommer Angabe vom Tarifrechner

EndeGueltigkeit	Pflicht	String	25	"2023-06-23T23:12:00+02:00	Gültigkeitsende des Tickets (Date-Time) oder Erstattungszeitpunkt in local time Format: UTC/GMT+1 (MEZ) im Winter UTC/GMT+2 (MESZ) im Sommer Angabe vom Tarifrechner
Verkaufszeitpunkt	Pflicht	String	19	02.08.2023 07:30:34	Zeitpunkt des Verkaufs z.B. 3.00 Uhr Nachts (Bestpreisoptimierung) oder Erstattungszeitpunkt in local time Format: „dd.mm.yyyy hh:mm:ss“ Angabe vom Lizenznehmer
TarifVersion	Pflicht	String	50	<6003_G220101_0000_E4_17>	Aktuelle Version des ProduktModuls des PV
VertriebsKanal	Optional	String	50	Fairtiq grün (oder Tickin, oder wasweissich)	Wird durch Whitelabelvergabe definiert, z.B. bwegt App, DB-App, VVSApp, etc.
Storno	Optional	String	1		S=Storno, F=Fehlbon (Maschinenstorno)
					identifiziert, ob und wenn ja um welche Art von Storno es sich handelt
StornoGrund	Pflicht wenn Storno gefüllt	String	30	CheckOut nicht korrekt erfasst	Weshalb wird storniert
Gutschrift	Optional	String	1	T	T = Tarifserverfehler Ggf. weitere
Erstattungsvorgangskennung	optional	String	30	Y143YZ Bad Cannstatt - Ulm	Der vom PV benannte Fehlercode und Hinweistext
Testdatensatz	Optional	Boolean	1		Als Möglichkeit, die Übermittlung und Verarbeitung zu testen. Der Datensatz wird dann in der Abrechnung am Ziel nicht berücksichtigt.

Beispiel:

```
curl --location 'https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/sale' \
--header 'x-api-key: WZaXTzjOju3OCIH3ZqL0DaRUrZ6Qd9F3a2UsStUw' \
--header 'Content-Type: application/json' \
--data '{
  "Header": "Verkauf",
  "KvpOrgID": 6235,
  "PvOrgID": 6234,
  "AbrechnungsID": "Abrechunungskopfidentifikation",
  "AbrechnungsPeriode": "08-2021",
  "Verkaufsliste": [
    {
      "AbrechnungsPositionID": 1,
      "KundenID": "123Test",
      "TicketTypID": "050301",
      "TicketTypDesc": "EinzelTagesTicket",
      "AbrechnungsPeriode": "08-2021",
      "PreisStufe": "1",
      "PreisEinzel": 532,
      "RaeumlicheGueltigkeit": "3-4-5",
      "BeginnGueltigkeit": "2021-08-10T12:23:42+02:00",
      "EndeGueltigkeit": "2021-08-11T07:00:00+02:00",
      "Verkaufszeitpunkt": "11.08.2021 03:00:00",
      "TarifVersion": "PVTM_6234_G200401_0000_E7_1_1.6_VVS",
      "VertriebsKanal": "1",
      "Storno": "S",
      "StornoGrund": "CheckOutnicht korrekt erfasst",
      "Gutschrift": "T",
      "Erstattungsvorgangskennung" : "Y143YZ Bad Canstatt - Ulm",
      "Testdatensatz": true
    },
    {
      "AbrechnungsPositionID": 2,
      "TicketTypID": "041201",
      "TicketTypDesc": "EinzelTicket",
      "AbrechnungsPeriode": "08-2021",
      "PreisEinzel": 632,
      "BeginnGueltigkeit": "2021-08-10T14:23:42+02:00",
      "EndeGueltigkeit": "2021-08-11T15:00:00+02:00",
      "Verkaufszeitpunkt": "17.08.2021 03:00:00",
      "TarifVersion": "PVTM_6234_G200401_0000_E7_1_1.6_VVS"
    }
  ]
}
'
```

Use Case 3: Fahrtketten-Datensätze:

Übermittlung der einzelnen Fahrtketten von Lizenznehmer zum territorial verantwortlichen Kundenvertragspartner. Nur für die Verkaufsdaten, nicht für Stornos.

Beispiel:

- Lizenznehmer betreibt App,
- Kunde fährt mit App im Verbundgebiet der VVS GmbH,
- App ermittelt die gefahrenen Fahrtketten

- Gefahrene und abgerechnete Fahrtkettendaten werden über die NVBW an die SSB AG übertragen, kein Kundendatenbezug.

Häufigkeit der Übertragung

mindestens einmal im Monat kumuliert

Daten

Kopf

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Header	Pflicht	String	30	Fahrtketten	Legt fest, welcher Nachrichtenart gesendet wird
Testdatensatz	Optional	Boolean		true, false	Möglichkeit die Übermittlung und Verarbeitung zu testen. Der Datensatz wird dann in der Abrechnung nicht berücksichtigt.
ApiVersion	ReadOnly	String	30	X.Y	Die Versionsnummer wird vom Microservice in den Datensatz geschrieben. Bereits existierende Werte werden überschrieben.
KvpOrgID	Pflicht	Integer	5		Org-ID nach VDV-KA des Lizenznehmers
PvOrgID	Pflicht	Integer	5		Org-ID des Verkehrsverbundes
AbrechnungsPeriode	Pflicht	String	7	08-2021	Monat für welchen das Ticket abgerechnet wurde, Format: mm-yyyy z.B. 08-2021
Ticketliste	Pflicht				JSON-Struktur mit Tickets

Tickets

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung

TicketTypID	Pflicht	String	20		ID des Tickettyps, z.B. 101 für bwtarif: ID_FKA
TicketTypDesc	Pflicht	String	100		Bezeichnung des Tickettyps, z.B.: Einzelticket Erwachsener für bwtarif: NAME_FKA (wie in der Statistik zur Verkaufsmeldung)
PreisStufe	Optional	String	20	3	Preisstufe des Produktes für bwtarif: leer
PreisEinzel	Pflicht	Integer	12	540	Preis für das Ticket in Euro-Cent ohne Trennzeichen für bwtarif: Einzelpreis nach tb_Preis bzw. aus Verkaufsmeldungen Umsatz/Anzahl
Fahrtenliste					JSON-Struktur mit Fahrten

Fahrten

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Haltestellenliste					JSON-Struktur mit Haltestellen
Liniendatenliste					JSON-Struktur der Liniendaten
Verkehrsmittel					JSON-Struktur des Verkehrsmittels

Haltestellen

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Ankunft	Optional	String	25	2023-06-23T23:12:00+02:00	Uhrzeit der Ankunft an einer Haltestelle in local time Format: UTC/GMT+1 (MEZ) im Winter UTC/GMT+2

					(MESZ) im Sommer Angabe vom Tarifrechner
Abfahrt	Optional	String	25	2023-06-23T23:12:00+02:00	Uhrzeit der Abfahrt an einer Haltestelle in local time Format: UTC/GMT+1 (MEZ) im Winter UTC/GMT+2 (MESZ) im Sommer Angabe vom Tarifrechner
Haltestelle					JSON-Struktur mit Haltestelle
Geoposition					JSON-Struktur der Geoposition

Haltestelle

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Nr	Pflicht	String	100 Länge kommt vom Tarifserver		DHID GlobalID
Name	Pflicht	String	100 Länge kommt vom Tarifserver		Name der Haltestelle

Geoposition

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Lat	Pflicht	Integer	10	48783333	Latitude
Lon	Pflicht	Integer	10	9183333	Longitude

Liniendaten

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung

Linie	Pflicht	String	100 Länge kommt vom Tarifserver		JSON-Struktur der Linie

Linie

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Nr	Pflicht	String	100 Länge kommt vom Tarifserver		Nummer der verwendeten Linie
Name	Pflicht	String	100 Länge kommt vom Tarifserver		Name der verwendeten Linie

Verkehrsmittel

Name "Feld im Tarifserver"	Key	Type	Länge	Value	Beschreibung
Nr	Pflicht	String	100 Länge kommt vom Tarifserver		Nummer des verwendeten Verkehrsmittels
Name	Pflicht	String	100 Länge kommt vom Tarifserver		Name des verwendeten Verkehrsmittels

Beispiel:

```
curl --location 'https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/drivechain'
\
--header 'x-api-key: WZaXTzjOju3OCIH3ZqL0DaRUrZ6Qd9F3a2UsStUw' \
--header 'Content-Type: application/json' \
--data '{
  "Header": "Fahrtketten",
  "Testdatensatz": true,
  "KvpOrgID": 6235,
  "PvOrgID": 6234,
  "AbrechnungsPeriode": "08-2021",
  "Ticketliste": [
    {
      "TicketTypID": "20151",
      "TicketTypDesc": "EinzelTicket Erwachsene",
      "PreisStufe": "Zone 32",
      "PreisEinzel": 255,
      "Fahrtenliste": [
        {
```

```
          "Verkehrsmittel": {
            "Nr": "13",
            "Name": "Zug (Nahverkehr)"
          },
          "Haltestellenliste": [
            {
              "Abfahrt": "2023-06-10T08:03:00+02:00",
              "Ankunft": "2023-06-10T08:03:20+02:00",
              "Haltestelle": {
                "Nr": "de:08127:20015",
                "Name": "Hessental Bahnhof"
              },
              "Geoposition": {
                "Lat": 49000000,
                "Lon": 9766681
              }
            },
            {
              "Abfahrt": "2023-06-10T08:04:00+02:00",
              "Ankunft": "2023-06-10T08:08:00+02:00",
              "Haltestelle": {
                "Nr": "de:08127:20013",
                "Name": "Schwäbisch Hall Bahnhof"
              },
              "Geoposition": {
                "Lat": 49000000,
                "Lon": 9733789
              }
            }
          ],
          "Liniendatenliste": [
            {
              "Linie": {
                "Nr": "de:nvbw:90T83_:",
                "Name": "RB 83"
              }
            }
          ]
        }
      ]
    }
  ]
}
'
```


Use Case 4: Einnahmesumme-Datensätze:

Übermittlung der Summe der Einnahmen je verkaufte Ticketart vom Lizenznehmer zum territorial verantwortlichen Kundenvertragspartner.
Ziel: Abrechnungsbasis zwischen Lizenznehmer und KVP/PV

Häufigkeit der Übertragung

mindestens einmal im Monat kumuliert

Daten

Kopf

Name	Key	Type	Länge	Value	Beschreibung

Header	Pflicht	String	30	Einnahmen	Legt fest, welcher Nachrichtenart gesendet wird
Testdatensatz	Optional	Boolean		true, false	Möglichkeit, die Übermittlung und Verarbeitung zu testen. Der Datensatz wird dann in der Abrechnung nicht berücksichtigt.
ApiVersion	ReadOnly	String	30	X.Y	Die Versionsnummer wird vom Microservice in den Datensatz geschrieben. Bereits existierende Werte werden überschrieben.
KvpOrgID	Pflicht	Integer	5		Org-ID nach VDV-KA des Lizenznehmers
PvOrgID	Pflicht	Integer	5		Org-ID des Verkehrsverbundes
AbrechnungsDatumAb	Pflicht	String	10	01.08.2021	Erster Tag der Abrechnungsperiode Format: dd.mm.yyyy
AbrechnungsDatumBis	Pflicht	String	10	10.08.2021	Letzter Tag der Abrechnungsperiode Format: dd.mm.yyy
AbrechnungsID	Pflicht	String	50	Af15054g	Eindeutige ID der Abrechnung
AbrechnungsPeriode	Optional	String	7	08-2021	Monat, für welchen das Ticket abgerechnet wurde, Format: mm-yyyy z.B. 08-2021
SummeAllerVerkäufe	Pflicht	Integer	12		Summierter Preis der Menge aller Tickets aus den Einnahmen in Cent ohne Komma, kann durch Stornos negativ werden
Einnahmenliste	Pflicht				JSON-Struktur der Einnahmen

Einnahmen

TicketTypID	Pflicht	String	20		Zeilenidentifier ist TicketID, z.B. „101“ oder Fehlercode für unbekannte Restposten (z.B. aus Tarifserverfehlerdifferenzen)

TicketTypDesc	Pflicht	String	100		Bezeichnung des Tickettyps, z.B.: Einzelticket Erwachsener
PreisStufe	Optional	String	20	3	Anzahl der Zonen
PreisEinzel	Pflicht	Integer	12	540	Preis für ein Ticket in Cent ohne Komma, immer postiver Wert
TarifVersion	Pflicht	String	50	PVTM_6234_G200401 _0000_E7_1_1.6_VVS	Aktuelle Version des ProduktModuls des PV
Menge	Pflicht	Integer	12		Menge der Tickets, kann bei Stornos negativ werden
PreisGesamt	Pflicht	Integer	12		Summierter Preis der Menge der Tickets in Cent ohne Komma, kann bei Stornos negativ werden

Beispiel:

```
curl --location 'https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/revenue' \
--header 'x-api-key: WZaXTzjOju3OCIH3ZqL0DaRUrZ6Qd9F3a2UsStUw' \
--header 'Content-Type: application/json' \
--data '{
  "Header": "Einnahmen",
  "KvpOrgID": 6235,
  "PvOrgID": 6234,
  "Tarifversion": "PVTM_6234_G200401_0000_E7_1_1.6_VVS",
  "AbrechnungsDatumAb": "01.08.2021",
  "AbrechnungsDatumBis": "10.08.2021",
  "AbrechnungsID": "10-08-2ss021",
  "AbrechnungsPeriode": "08-2022",
  "SummeAllerVerkaeufe": 164376,
  "VertriebsKanal": "SSB App",
  "Einnahmenliste": [
    {
      "TicketTypID": "50254",
      "TicketTypDesc": "EinzelTagesTicket Online",
      "PreisStufe": "4",
      "PreisEinzel": 1044,
      "TarifVersion": "PVTM_6234_G200401 _0000_E7_1_1.6_VVS",
      "Menge": 103,
      "PreisGesamt": 107532
    },
    {
      "TicketTypID": "50808",
      "TicketTypDesc": "StadtTicket EinzelTagesTicket",
      "PreisEinzel": 300,
      "TarifVersion": "PVTM_6234_G200401 _0000_E7_1_1.6_VVS",
      "Menge": 99,
      "PreisGesamt": 29700
    },
    {
      "TicketTypID": "20154",
      "TicketTypDesc": "EinzelTicket Mobil",
      "PreisStufe": "4",
      "PreisEinzel": 522,
      "TarifVersion": "PVTM_6234_G200401 _0000_E7_1_1.6_VVS",
      "Menge": 52,
      "PreisGesamt": 27144
    }
  ]
}
'
```

Technische Datenübermittlung

a) Vom Lizenznehmer (App-Betreiber) an Microservice (NVBW)

Endpoint Testsystem

Endpoint-URL	https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test
API-Schlüssel	WZaXTzjOju3OCIH3ZqL0DaRUrZ6Qd9F3a2UsStUw
Content-Type	application/json

Die Endpoint-URL zur Ansprache des Microservice bezieht sich auf das Testsystem im Hosting bei Amazon AWS.

Die UseCases 1-4 werden erreicht unter:

```
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/customer
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/sale
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/drivechain
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/revenue
```

und der Server Heartbeat mit:

```
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/test/ping
```

Endpoint Produktivsystem

Endpoint-URL	https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod
API-Schlüssel	werden gesondert mitgeteilt
Content-Type	application/json

Die UseCases 1-4 werden erreicht unter:

```
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod/customer
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod/sale
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod/drivechain
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod/revenue
```

und der Server Heartbeat mit:

```
https://0aaflvt0c6.execute-api.eu-central-1.amazonaws.com/prod/ping
```

Das Testsystem und das Produktivsystem mit dem Microservice ist nur noch für bekannte IP-Adressbereiche zugänglich. Die Lizenznehmer müssen NVBW daher die IP-Adresse/IP-Adressbereich mitteilen, von dem aus das Testsystem bzw. das Produktivsystem angefragt wird. Im Auftrag von NVBW werde diese in der firewall hinterlegt. **Zur Freischaltung werden statische externe IP-Adressen benötigt.** Das sind typischerweise die IP-Adressen, die man zum Beispiel im Browser mit https://www.heise.de/netze/tools/meine-ip-adresse/ oder besser auf der Kommandozeile z.B. mit curl ipinfo.io/ip ermitteln kann. Bei iv6 ist extern und intern identisch.

Datenformat: JSON[1]

Pro PvOrgID im Header muss gebündelt ein separater Datensatz gesendet werden, Z.B. alle Kundenstammdaten der Periode. Keinesfalls soll jeder Kundendatensatz separat versendet werden.

Übertragungs-Limit

Die maximale Größe des JSON-Payloads Im Body einer Anfrage beträgt 6MB. Eine Überschreitung dieses Limits führt zu einer Ablehnung der Anfrage.
Bei Anfragen mit Payloads die größer sind als das Limit, empfiehlt sich die Zerlegung in mehrere Anfragen.

b) Konfiguration des Microservice

Konfiguration der Ziele und Ausgabeformate

Konfiguration der Ziele und der Ausgabeformate wird mit einer yaml-Datei beschreiben, die auf dem Microservice-Server abgelegt wird.

Die yaml-Datei wird vom Betreiber des Microservices (NVBW) nach den Wünschen der tKVP/PV erstellt werden (alternativ: Lieferung einer Beispieldatei durch NVBW und Anpassung und Rücklieferung durch die tKVP).

Ausgangsformate

Während für das Eingangsformat des Microservices stets JSON gefordert ist, lässt das Ausgangformat folgende drei Stammformate zu:

- XML
- JSON
- CSV

Auf Basis dieser drei Stammformate lassen sich dann je Ziel-PV/tKVP das vom Ziel gewünschte Format in der yaml-Datei konfigurieren.

Datenstruktur

Datensätze werden vom Microservice pro PvOrgID getrennt ausgegeben.

[1] Die angegebenen Schlüssel in den definierten Strukturen sind nicht case sensitive.

Pro eingehender Datensatz, wird ein Datensatz ausgegeben. Die Datenstruktur der Eingangsfiles wird im Ausgang nicht zusammengeführt. Das Zielformat und die Reihenfolge der Daten, Feldnamen und die Anzahl der Felder entspricht in der Regel der Eingangsstruktur.

Erfolgt eine Ausgabe im cvs-Format, so lässt sich das gewünschte Zielformat beeinflussen. Die Reihenfolge, Feldnamen oder Anzahl der Felder kann pro PvOrgID separat festgelegt werden.
Sollte die Reihenfolge, Feldnamen oder Anzahl nach Go-Live geändert werden müssen, ist eine Frist von 3 Monaten zur Produktivsetzung der benötigten Änderung einzuhalten. Produktivsetzungstermin wäre jeweils zum Quartalsende.

Häufigkeit der Datenübertragung

Die Häufigkeit der Datenübertragung richtet sich nach der Häufigkeit der Eingangsnachrichten.

c) Anforderung an Datenabnehmer/Ziel

Ausgabe der Daten vom Microservice auf gesichertes Ziellaufwerk des tKVP/PV per SFTP. SFTP basiert auf dem SSH-Protokoll. Die Authentifizierung erfolgt mit einem Zertifikat. Der öffentliche Schlüssel muss auf dem SFTP-Server am Ziel eingespielt werden. Mit Hilfe des privaten Schlüssels erfolgt vom Microservice die Client-Authentifizierung zum Datei-Upload.

Der Abnehmer muss hierzu einen SFTP-Server bereitstellen. Weiter wird zum Auslesen der Daten ein SFTP-Client vom Datenabnehmer benötigt.
Ein bestimmtes Produkt für den SFTP-Server wird nicht empfohlen; auch nicht eine bestimmte Größe der Festplatte. Info: Die einzelnen Datensätze sind relativ klein im Bereich von wenigen kByte. Die Anzahl hängt von der Nutzung des CiCoBW-Systems ab. Insgesamt rechnen wird nach heutigen Maßstäben mit geringem Platzbedarf. Im Übrigen soll der SFTP-Server am Ziel nur als Austauschplattform eingesetzt werden und typischerweise nicht zur Aufbewahrung der Daten dienen.

Der Abnehmer ist für Datensicherheit und Datenschutz
auf dem Ziellaufwerk verantwortlich (Sicherung mit Benutzernamen und Zertifikat z. B.).

Die Dateibezeichnungsstruktur wird vom Ziel-KVP/PV vorgegeben, z.B.:

Ab Version 2.2 können optional die PvOrgID und KvpOrgID über die Platzhalter `${PvOrgID}` und `${KvpOrgID}` zusätzlich zum Dateinamen in den Dateibezeichner integriert werden.

CiCoBW_“HeaderTyp“__${PvOrgID}_${KvpOrgID}_Verkaufsdaten.csv

Bsp. :
CiCoBW_KVSH_6003_1234_Verkaufsdaten_20260101-120000000.csv

Die Datei beinhaltet CiCoBW-Daten zum Use case Verkaufsdaten an die Tariforganisation mit Header KVSH und PVOrg_ID 6003. Sie kommen vom Lizenznehmer mit KVP_OrgID 1234 und wurden am 01.Januar 2026 um 12 Uhr vom Microservice versandt.

Der Header-Typ sollte typischerweise ein Kürzel für die Tariforganisation beinhalten. Er muss manuell von der NVBW konfiguriert werden und ist dann fix.

Die PvOrgID der Ziele übernimmt der Microservice dynamisch aus den Inhalten der Use Cases. Sollten auf einem Zielserver also Daten für unterschiedliche Verbünde lagern, so lassen sie sich anhand der Dateinamen unterscheiden, für wen diese bestimmt sind.

Die KVPOrgID der Datenherkunft (Lizenznehmer) übernimmt der Microservice dynamisch aus den Inhalten der Use Cases. Dadurch lassen sich auf einem Zielserver die Daten unterschiedlicher Lizenznehmer bereits am Dateinamen voneinander unterscheiden.

Am Ende des Dateinamens hängt der Microservice noch einen Zeitstempel (Datum-Uhrzeit) an.

Es wird für jeden Use Case je eine separate Datei erzeugt.

Die weitere Verarbeitung der Dateien liegt im Verantwortungsbereich des tKVP/PV auf Datenabnehmerseite.

Der Abnehmer hat die Daten des Zielservers an NVBW zu übermitteln:

- URL
- IP
- Port
- Benutzername

Am Benutzernamen wird das öffentliche Zertifikat der NVBW zur Authentisierung hinterlegt.
Ein Kennwort wird nicht benutzt.

Es ist ein Zugriff für den Microservice einzurichten. Sofern der Ziel-SFTP-Server den IP-Range auf bestimmte Zugriffe eingrenzen will, kann die IP des Microservice-Testservers und später des Produktivservers an tKVP/PV übermittelt werden.

Die folgenden „Outgoing IP-Adressen“ verwendet der Microservice:

- 3.69.67.191
- 3.126.238.171

d) Fehlerhandling

Info an Lizenznehmer:

- Empfang am Microservice gescheitert
- Verarbeitung am Microservice gescheitert
    - Validitätsprüfung mit Rückgabe des ersten Feldes, welches fehlerhaft war
        - Formatfehler (Datentyp)
        - Feldlängenprüfung
        - Pflichtfelder
    - PvOrgID nicht gefunden
    - Header Parameter falsch
- Zustellung im Zielsystem gescheitert

Lizenznehmer muss eine entsprechende E-Mail-Adresse angeben, an welche die Info verschickt werden muss. Info an KVP/PV:

- Ist das Ziellaufwerk nicht erreichbar, muss eine sprechende Fehlermeldung an den KVP/PV per E-Mail geschickt werden
- KVP/PV muss eine entsprechende E-Mail-Adresse angeben, an welche die Info verschickt werden muss.

Was muss KVP/PV bzw. Lizenznehmer im Fehlerfall tun? Das Melde-System informiert nur, greift aber nicht korrigierend ein. Die Verantwortung zur Fehlerbehebung liegt auf Seiten der Lizenznehmer bzw. KVP/PV.
Ebenso notwendig ist die Bestätigung bei erfolgreicher Zustellung, automatisiert verarbeitbar durch Lizenznehmersystem.

Testdatensatz

In allen Use Cases wurde das optionale Feld „Testdatensatz“ als Boolean aufgenommen (default=FALSE). Dieser bietet die Möglichkeit, auch in einer Produktivumgebung einen Datensatz als Testfall als solchen zu kennzeichnen (TRUE), damit dieser vom den tKVP/PV nicht in der Abrechnung berücksichtigt wird. Auf Lieferantenseite ist das Feld optional, auf Abnehmerseite hingeben muss es beachtet werden.

API-Version

Der Microservice fügt automatisch das Attribut „APIVersion“ vom Typ ReadOnly nach dem Header ein. Die Versionsnummer wird vom Microservice in den Datensatz geschrieben.
Bereits existierende Werte werden überschrieben. Die APIVersion wird durch Datagroup/NVBW festgelegt werden und wird sich bei strukturellen Änderungen an den Daten erhöhen.
Der Initialwert ist „1.0“. Dies bedeutet nicht, dass der Microservice in mehreren Versionen gleichzeitig verfügbar ist. Das Feld soll lediglich zur Information für die Abnehmer/Zielsysteme dienen.