



Bundesverband für Energie- und
Wasserdatenmanagement e.V.

ERP-Implementierungsleitfaden

On-Site Roles API v2.0

Version 2.0.3

Oktober 2026 | Zielgruppe: ERP-Anbieter / Softwareentwickler

ERP-Implementierungsleitfaden – bved On-Site Roles API v2.0

Version: 2.0.3

Datum: 2026-10-01

Hinweis: Dieser Leitfaden wurde mit Hilfe von KI erzeugt.

Zielgruppe: ERP-Anbieter / Softwareentwickler (Wohnungswirtschaft)

Spezifikation: bved On-Site Roles API v2.0.3 (OpenAPI: `bved-on-site-roles-2-0.openapi.yaml`)

Inhalt

1. Einführung und Kontext
 2. Voraussetzungen
 3. Authentifizierung
 4. API-Endpunkte
 5. Vollständige Feldreferenz
 6. ERP-Datenmapping
 7. Implementierungsmuster
 8. Vollständige JSON-Beispiele
 9. Fehlerbehandlung
 10. Logging-Empfehlungen
 11. Implementierungs-Checkliste
 12. Testempfehlungen
 13. Anhang A: Referenz zu Version 3.10
-

1. Einführung und Kontext

Was ist die On-Site Roles API?

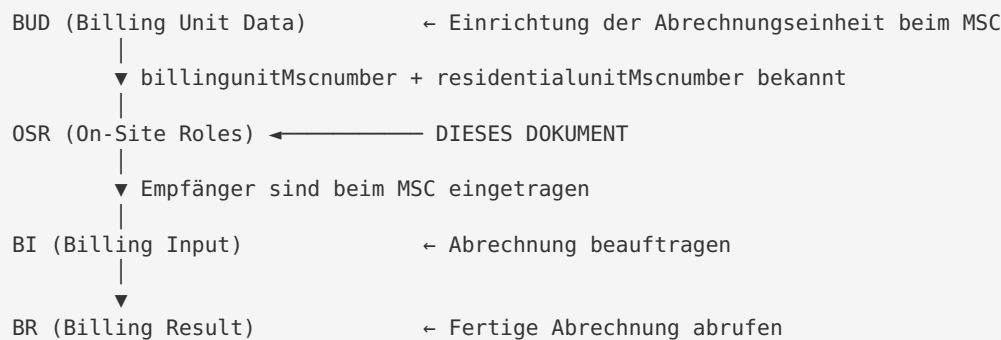
Die **bved On-Site Roles API v2.0** ist die REST-Schnittstelle, über die eine Hausverwaltung (Property Manager, PM) die aktuellen Mieter-, Eigentümer- und Empfängerinformationen für jede Nutzeinheit (NE) an den zuständigen Messdienstleister (MSC) meldet.

Übertragen werden:

- **Partner** — natürliche und juristische Personen (Mieter, Eigentümer, Verwalter), die eine Rolle in einer Nutzeinheit einnehmen

- **Abrechnungsempfänger** (`billingcontracts`) – wer erhält die Heizkostenabrechnung, für welchen Zeitraum?
- **uVI-Empfänger** (`consumptioninformationcontracts`) – wer erhält die unterjährige Verbrauchsinformation (uVI) gemäß EED-Richtlinie?

Einordnung in die bved-API-Familie



Abhängigkeit: OSR setzt voraus, dass die Abrechnungseinheit und ihre Nutzeinheiten in der BUD-API beim MSC bereits angelegt sind (`billingunitMscnumber` und `residentialunitMscnumber` müssen bekannt sein).

Kernprinzip: Vollständige Datenlieferung je Nutzeinheit

Jeder OSR-POST enthält den **vollständigen aktuellen Stand** einer Nutzeinheit – keine Deltas, keine Änderungsmeldungen. Der MSC:

- **Ergänzt** bisher unbekannte Rollen
- **Aktualisiert** bekannte Rollen
- **Beendet oder löscht** nicht mehr gesendete aktive Rollen (MSC-spezifisch)
- **Lässt historische Daten unverändert** (bereits beendete Verträge, die nicht mehr gesendet werden)

2. Voraussetzungen

Voraussetzung	Details
OAuth 2.0 Credentials	Client ID und Client Secret vom MSC – nötig für Token-Abruf
MSC-Nummern	<code>billingunitMscnumber</code> (9 Stellen) und <code>residentialunitMscnumber</code> (4 Stellen) müssen für jede zu meldende NE bekannt sein (aus BUD-Setup)

Voraussetzung	Details
Vertragsvereinbarung	Eine vertragliche Vereinbarung mit dem MSC ist Voraussetzung; sie umfasst insbesondere die Datenschutz-/Auftragsverarbeitungsvereinbarung (AVV) für die Übertragung personenbezogener Daten. Fehlt die erforderliche Berechtigung, antwortet der MSC mit 403
HTTPS/TLS	Alle API-Aufrufe ausschließlich über verschlüsselte Verbindung
Basis-URL	Wird vom jeweiligen MSC bereitgestellt — kein gemeinsamer Endpunkt, jeder MSC hat seine eigene URL

3. Authentifizierung

Die API ist mit **OAuth 2.0 Client Credentials Flow** gesichert.

Token-Abruf

```
POST {tokenUrl}
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id={client_id}
&client_secret={client_secret}
```

Die **tokenUrl** wird vom MSC mitgeteilt.

Token-Verwendung

```
POST /onsiteroles/v2/billingunits/123456789/residentialunits/0010/on-site-roles
Authorization: Bearer {access_token}
Content-Type: application/json
```

Implementierungshinweise

- Access Token vor Ablauf erneuern (TTL ist MSC-spezifisch — typisch 3600 Sekunden)
- Token serverseitig cachen, nicht per Request neu abrufen
- Bei **401** -Antwort: Token erneuern und Request einmalig wiederholen
- Credentials nicht im Quellcode oder in Logs speichern
- Der gleiche Token kann für alle bved-APIs desselben MSC wiederverwendet werden, sofern der MSC das so vorsieht

4. API-Endpunkte

Basis-URL: `{msc-base-url}/onsiteroles/v2`

Alle Antworten enthalten den Response-Header `X-API-VERSION` mit der vom MSC implementierten Version.

4.1 GET — Beim MSC gespeicherte Daten lesen

```
GET /billingunits/{billingunit}/residentialunits/{residentialunit}/on-site-roles
```

Die hier gelieferten Daten enthalten die Rollen und Partner, die dem MSC vorliegen. Es werden die zum Zeitpunkt der Abfrage vorliegenden Daten zur Nutzeinheit vom MSC übermittelt. Abhängig vom MSC können historische Daten abweichend von der letzten Übermittlung zurückgegeben werden. Diese entsprechen z.B. der Abrechnungshistorie des MSC und sind für die aktuelle Periode nicht relevant.

Verwendungszwecke:

- Abgleich vor Korrektur
- Debugging / Support
- Prüfung vor Abrechnungsbeauftragung

Wichtig: GET wird nicht von allen MSC unterstützt. `501 Not Implemented` ist eine reguläre Antwort und kein Fehler. Das ERP muss `501` explizit abfangen und darf es nicht als Ausfall behandeln.

Antwort bei Erfolg: `200 OK`

```
{
  "billingunitMscnumber": "123456789",
  "billingunitPmnumber": "WEG-2024-001",
  "residentialunitMscnumber": "0010",
  "residentialunitPmnumber": "NE-Erdgeschoss-Links",
  "lastupdate": "2025-01-15T10:30:00Z",
  "partners": [ ... ],
  "billingcontracts": [ ... ],
  "consumptioninformationcontracts": [ ... ]
}
```

4.2 POST — Daten senden / aktualisieren

```
POST /billingunits/{billingunit}/residentialunits/{residentialunit}/on-site-roles
Content-Type: application/json
```

Sendet den vollständigen aktuellen Rollendatensatz für eine Nutzeinheit.

Antwort bei Erfolg: `200 OK`

```
{
  "transactionid": "T-0000042"
}
```

Die `transactionid` identifiziert den Vorgang eindeutig beim MSC und ist für Support-Anfragen essenziell.

Antwort bei Validierungsfehlern: 400 Bad Request

```
{
  "transactionid": "T-0000099",
  "messages": [
    {
      "type": "ERROR",
      "shortmessage": "MISSING",
      "message": "Partner 'P-4712' in billingrecipient referenziert, aber nicht in partners[] enthalten",
      "language": "de"
    }
  ]
}
```

Jede Meldung ist ein `ValidationMessage` mit `type` (`ERROR` oder `WARNING`), `shortmessage`, `message` und optional `language`. Das ERP muss **beide** `type`-Werte verarbeiten können. Die Erfolgsantwort (`200`) enthält keine Meldungen – sie besteht nur aus der `transactionid`.

4.3 Path-Parameter

Parameter	Feld im Body	Typ	Länge	Beispiel
<code>billingunit</code>	<code>billingunitMscnumber</code>	string	9 Stellen	123456789
<code>residentialunit</code>	<code>residentialunitMscnumber</code>	string	4 Stellen	0010

Path-Parameter und Body-Felder müssen konsistent sein.

5. Vollständige Feldreferenz

5.1 SendOnSiteRoles (POST Request Body)

Feld	Typ	Pflicht	Länge / Format	Beschreibung
<code>billingunitMscnumber</code>	string	Ja	9 Stellen	MSC-Nummer der Abrechnungseinheit

Feld	Typ	Pflicht	Länge / Format	Beschreibung
billingunitPmnumber	string	Nein	1–15 Zeichen	ERP-eigene Nummer der AE (Referenz für MSC)
residentialunitMscnumber	string	Ja	4 Stellen	MSC-Nummer der Nutzeinheit
residentialunitPmnumber	string	Nein	1–20 Zeichen	ERP-eigene Nummer der NE
partners	array	Ja	—	Liste aller im Request referenzierten Partner
billingcontracts	array	Nein	—	Abrechnungsempfänger-Verträge der NE
terminateallbillingcontracts	boolean	Nein	—	true → alle laufenden BC beim MSC beenden
terminateallbillingcontractsat	date	Nein	YYYY-MM-DD	Enddatum bei Massenbeendigung
consumptioninformationcontracts	array	Nein	—	uVI-Empfänger (unterjährige Verbrauchsinformation)
terminateallconsumptioninformationcontracts	boolean	Nein	—	true → alle laufenden CIC beim MSC beenden
terminateallconsumptioninformationcontractsat	date	Nein	YYYY-MM-DD	Enddatum bei Massenbeendigung

5.2 OnSitePartner

Feld	Typ	Pflicht	Länge / Format	Beschreibung
partnerPmnumber	string	Ja	1–20 Zeichen	Eindeutige ERP-ID des Partners — Schlüssel für Vertragsreferenzen
name	string	Ja	1–35 Zeichen	Nachname (Privatperson) oder Firmenname
commercial	string	Ja	Enum	COMMERCIAL / NON_COMMERCIAL / UNKNOWN

Feld	Typ	Pflicht	Länge / Format	Beschreibung
firstname	string	Nein	1–35 Zeichen	Vorname (nur natürliche Personen)
additionalname	string	Nein	1–35 Zeichen	Namenszusatz (nicht alle MSC unterstützen dieses Feld)
email	string	Nein	1–241 Zeichen	E-Mail-Adresse (Datenschutzvereinbarung beachten)
phonenumbers	array	Nein	—	Liste von Telefonnummern
alternativeaddress	Address	Nein	—	Abweichende Postadresse — wird nur beim Hauptempfänger für den Versand genutzt (siehe §5.3)
taxnumber	string	Nein	1–16 Zeichen	Steuernummer (z.B. bei gewerblichen Rollen)
bankaccount	BankAccount	Nein	—	IBAN des Partners

phonenumbers[] — Phone-Objekt:

Feld	Typ	Pflicht	Gültige Werte	Beschreibung
number	string	Ja	1–40 Zeichen; nur 0-9 + () / - und Leerzeichen	Telefonnummer
type	string (enum)	Ja	MOBILE , LANDLINE , PRIVATE , BUSINESS , OTHER	Nummerntyp

alternativeaddress — Address-Objekt:

Feld	Typ	Pflicht	Länge	Beschreibung
street	string	Nein	1–35 Zeichen	Straße und Hausnummer
postalcode	string	Nein	1–10 Zeichen	Postleitzahl
city	string	Nein	1–35 Zeichen	Stadt
country	string	Nein	genau 3 Zeichen	ISO 3166-1 alpha-3 (z.B. DEU , AUT , CHE)

bankaccount — BankAccount-Objekt:

Feld	Typ	Pflicht	Länge	Beschreibung
<code>iban</code>	string	Ja	15–32 Zeichen	IBAN ohne Leerzeichen (z.B. <code>DE91100000000123456789</code>)

5.3 BillingContract

Feld	Typ	Pflicht	Format	Beschreibung
<code>validfrom</code>	date	Ja	YYYY-MM-DD	Vertragsbeginn / Einzugsdatum
<code>validto</code>	date	Nein	YYYY-MM-DD	Vertragsende / Auszugsdatum (leer = laufend)
<code>vacancy</code>	boolean	Ja	—	<code>true</code> = Leerstand; <code>false</code> = bewohnte Einheit
<code>billingrecipient</code>	BillingRecipient	Ja	—	Hauptempfänger der Abrechnung – zugleich der postalische Empfänger
<code>additionalbillingrecipient</code>	BillingRecipient	Nein	—	Zusätzlicher Empfänger (z.B. bei WEG) – wird nur auf der Abrechnung genannt, seine Adresse wird nicht verwendet
<code>sender</code>	BillingSender	Nein	—	Alternativer Auftraggeber / Leistungsgeber

`billingrecipient` / `additionalbillingrecipient` – BillingRecipient-Objekt:

Feld	Typ	Pflicht	Beschreibung
<code>owner</code>	boolean	Ja	<code>true</code> = Eigentümer (WEG-Adressierung); hat keinen Einfluss auf Abrechnung
<code>partnerPmnumber</code>	string	Ja	Referenz auf <code>partnerPmnumber</code> in <code>partners[]</code>

Postalische Adressierung: Der `billingrecipient` ist der **Hauptabrechnungsempfänger und zugleich der postalische Empfänger** – die Abrechnung wird an ihn versandt, an seine `alternativeaddress` bzw. andernfalls an die Adresse der Nutzeinheit. Der `additionalbillingrecipient` wird lediglich **zusätzlich auf der Abrechnung genannt**; seine Adresse wird für den Versand **nicht** verwendet. Eine `alternativeaddress` am zusätzlichen

Empfänger bleibt damit wirkungslos. Sollen zwei Personen die Abrechnung jeweils an ihre eigene Adresse erhalten, ist das über `additionalbillingrecipient` nicht abbildbar.

sender — BillingSender-Objekt:

Feld	Typ	Pflicht	Beschreibung
<code>partnerPmnumber</code>	string	Ja	Referenz auf <code>partnerPmnumber</code> in <code>partners[]</code>

5.4 ConsumptionInformationContract

Feld	Typ	Pflicht	Format	Beschreibung
<code>validfrom</code>	date	Ja	YYYY-MM-DD	Beginn des Empfangs von Verbrauchsinformationen
<code>validto</code>	date	Nein	YYYY-MM-DD	Ende des Empfangs (leer = laufend)
<code>partnerPmnumber</code>	string	Ja	—	Referenz auf <code>partnerPmnumber</code> in <code>partners[]</code>

5.5 Terminierungsflags — Regeln

Szenario	<code>terminateallbillingcontracts</code>	<code>billingcontracts</code>	Ergebnis
Normale Meldung	<code>false</code> (oder fehlt)	gefüllt	Normales Update
Keine Änderung an BC	<code>false</code> (oder fehlt)	leer / fehlt	BC beim MSC unverändert
Alle BC beenden	<code>true</code>	leer	MSC beendet alle laufenden BC
Ungültig	<code>true</code>	gefüllt	400 -Fehler

Gleiches gilt analog für `terminateallconsumptioninformationcontracts`.

6. ERP-Datenmapping

6.1 ERP-Stammdaten → `partners[]`

Aus dem ERP werden typischerweise folgende Entitäten auf `OnSitePartner`-Objekte gemappt:

ERP-Entität	name	firstname	commercial	Hinweise
Natürliche Person (Mieter/Eigentümer)	Nachname	Vorname	NON_COMMERCIAL	
GmbH, AG, Genossenschaft	Firmenname	—	COMMERCIAL	firstname weglassen
Einzelunternehmer	Firmenname	—	COMMERCIAL	
Hausverwaltung (eigene Firma)	Firmenname	—	COMMERCIAL oder NON_COMMERCIAL (im Fall von Privatpersonen)	Für Leerstand- Empfänger

partnerPmnumber sollte der unveränderliche Primärschlüssel des Mieters/Eigentümers im ERP sein (nicht die Vertragsnummer, die sich bei Mieterwechsel ändert). Die **partnerPmnumber** muss innerhalb der **partners[]** -Liste eines Requests eindeutig sein. Andernfalls sind die Referenzen aus **billingcontracts** / **consumptioninformationcontracts** nicht auflösbar und der Request wird mit **400** abgelehnt.

6.2 Abrechnungsempfänger → **billingcontracts[]**

ERP-Feld	API-Feld	Besonderheit
Einzugsdatum	validfrom	Pflicht – immer befüllen
Auszugsdatum	validto	Leer lassen wenn Vertrag noch läuft
Empfänger	billingrecipient.partnerPmnumber	Referenz auf Mieter- bzw. Eigentümer-Partner; dieser Partner erhält die Abrechnung postalisch
Ist Eigentümer	billingrecipient.owner	true bei WEG-Eigentümer
Wohnung leer	vacancy: true	Immer melden, auch Leerstand braucht Empfänger
Zusätzlicher Empfänger	additionalbillingrecipient	Steht zusätzlich mit auf der Abrechnung; seine Adresse wird für den Versand nicht genutzt
Leistungsgeber abweichend	sender.partnerPmnumber	Nur wenn Auftraggeber ≠ Empfänger

6.3 uVI-Empfänger → `consumptioninformationcontracts[]`

ERP-Feld	API-Feld	Hinweis
Beginn uVI-Empfang	<code>validfrom</code>	Pflicht
Ende uVI-Empfang	<code>validto</code>	Leer = laufend
Empfänger	<code>partnerPmnumber</code>	Muss auch in <code>partners[]</code> enthalten sein

Pflicht zur Trennung: Billing-Empfänger und uVI-Empfänger müssen in **separaten Requests** übermittelt werden. Die Spezifikation schreibt das explizit vor, damit Validierungsfehler eines Dienstleistungstyps nicht den anderen blockieren. Ein Request enthält entweder `billingcontracts` oder `consumptioninformationcontracts` – niemals beide befüllt gleichzeitig.

6.4 Vollständigkeit der `partners[]` -Liste

Kritische Regel: Jede `partnerPmnumber`, die in einem `billingcontract`, `consumptioninformationcontract` oder einem `sender`-Feld referenziert wird, **muss** in `partners[]` enthalten sein. Andernfalls antwortet der MSC mit `400` / `shortmessage: MISSING`.

Konstruktionsempfehlung:

1. Alle `partners[]` mit `partnerPmnumber` sammeln
2. Alle Verträge (BC, CIC) aufbauen, Referenzen aus `partners[]` verwenden
3. Vor dem Senden: alle `partnerPmnumber` aus Verträgen gegen `partners[]` prüfen

7. Implementierungsmuster

7.1 Grundregel: Getrennte Requests je Dienstleistungstyp

Die Spezifikation schreibt vor: Abrechnungsempfänger (`billingcontracts`) und uVI-Empfänger (`consumptioninformationcontracts`) werden in **separaten Requests** übermittelt. Beide befüllt im gleichen Request ist ein Fehler.

Das Prinzip gilt für alle Szenarien – Ersterfassung, Mieterwechsel, Beendigung:

Trigger: Mieterwechsel mit neuem uVI-Empfänger
 → Request 1: `billingcontracts` befüllt, `consumptioninformationcontracts` leer
 → Request 2: `billingcontracts` leer, `consumptioninformationcontracts` befüllt

Ein leeres Array mit `terminateall*: false` (oder fehlendes Feld) bewirkt keine Änderung – das ist der korrekte Weg, nur einen Typ zu aktualisieren.

7.2 Wann senden?

Trigger-Ereignis im ERP	Empfehlung
Neuer Abrechnungsempfänger gespeichert	Sofort nach Speichern
Einzugsdatum geändert	Sofort nach Speichern
Auszugsdatum eingetragen	Sofort nach Speichern
Empfängeradresse geändert	Sofort nach Speichern
Namensänderung eines Partners (z.B. durch Heirat)	Sofort nach Speichern
Leerstand eingetragen	Sofort nach Speichern
Ersteinrichtung neue Liegenschaft	Nach BUD-Setup für alle NE
Batch-Import aus Fremdsystem	Innerhalb 1 Stunde nach Import

7.3 Muster 1: Ersteinrichtung einer neuen Nutzeinheit

NE 0030 wird erstmals eingerichtet. Erster Mieter A zieht am 2025-01-01 ein.

POST (Billing):

```
partners: [Partner A]
billingcontracts: [{ validfrom: 2025-01-01, billingrecipient: A }]
terminateallbillingcontracts: false
```

7.4 Muster 2: Mieterwechsel – vollständige Nutzeinheit senden

NE 0010 hat folgende Geschichte im ERP:
 Vormieter X: 2023-01-01 bis 2023-12-31
 Altmmieter A: 2024-01-01 bis 2024-12-31
 Neummieter B: ab 2025-01-01 (neu eingezogen)

→ Es werden ALLE drei Verträge gesendet, auch X der sich nicht ändert.

POST (Billing):

```
partners: [Partner X, Partner A, Partner B] ← alle referenzierten Partner
billingcontracts: [
  { validfrom: 2023-01-01, validto: 2023-12-31, billingrecipient: X },
  { validfrom: 2024-01-01, validto: 2024-12-31, billingrecipient: A },
  { validfrom: 2025-01-01, billingrecipient: B }
]
terminateallbillingcontracts: false
```

7.5 Muster 3: Mieterwechsel mit uVI – zwei separate Requests

Gleiche NE 0010, Neumieter B erhält auch die unterjährige Verbrauchsinformation (uVI).

Request 1 – Abrechnungsempfänger (vollständige NE):

```
partners: [Partner X, Partner A, Partner B]
billingcontracts: [alle drei Verträge wie oben]
terminateallbillingcontracts: false
```

Request 2 – uVI-Empfänger (vollständige NE):

```
partners: [Partner A, Partner B] ← nur die in CIC referenzierten Partner
consumptioninformationcontracts: [
  { validfrom: 2024-01-01, validto: 2024-12-31, partnerPmnumber: A },
  { validfrom: 2025-01-01, partnerPmnumber: B }
]
terminateallconsumptioninformationcontracts: false
```

Partner müssen in jedem Request separat mitgeliefert werden – nur die, die in diesem Request referenziert werden.

7.6 Muster 4: Leerstand (vollständige Nutzeinheit senden)

NE 0010: Nach Auszug von Neumieter B steht die Wohnung ab 2025-06-01 leer.

POST (Billing):

```
partners: [Partner X, Partner A, Partner B, Partner Verwaltung]
billingcontracts: [
  { validfrom: 2023-01-01, validto: 2023-12-31, billingrecipient: X },
  { validfrom: 2024-01-01, validto: 2024-12-31, billingrecipient: A },
  { validfrom: 2025-01-01, validto: 2025-05-31, billingrecipient: B },
  { validfrom: 2025-06-01, vacancy: true, billingrecipient: Verwaltung }
]
terminateallbillingcontracts: false
```

Leerstand muss immer explizit mit `vacancy: true` gemeldet werden.

7.7 Muster 5: Alle Verträge beenden – zwei separate Requests

Liegenschaft wird aufgelöst, alle Verträge zum 2025-12-31 beenden.

Request 1 – Alle Abrechnungsempfänger beenden:

```
partners: []
billingcontracts: []
terminateallbillingcontracts: true
terminateallbillingcontractsat: "2025-12-31"
```

Request 2 – Alle uVI-Empfänger beenden:

```
partners: []
consumptioninformationcontracts: []
terminateallconsumptioninformationcontracts: true
terminateallconsumptioninformationcontractsat: "2025-12-31"
```

7.8 Idempotenz nutzen

POST `/on-site-roles` ist by design idempotent: Dasselbe Request erneut senden hat keine unerwünschten Nebeneffekte. Dieses Verhalten ermöglicht sichere Retry-Logik:

1. Request-Body lokal puffern
2. POST senden
3. Netzwerkfehler / Timeout → identischen Body direkt erneut senden
4. Erst nach 200-Antwort als erfolgreich markieren
5. Keine kumulative Zustandsverwaltung nötig

8. Vollständige JSON-Beispiele

Zwei Grundregeln für alle Beispiele:

1. `billingcontracts` und `consumptioninformationcontracts` werden immer in **separaten Requests** übermittelt.
2. Jeder Request enthält den **vollständigen Datensatz der Nutzeinheit** — alle Verträge, nicht nur die geänderten. Partner müssen für jeden referenzierten Vertrag mitgeliefert werden.

Die Beispiele 8.2–8.4 verwenden eine gemeinsame Nutzeinheit (NE 0010) mit wachsender Geschichte:

Zeitraum	Mieter	Partner-ID
2023-01-01 – 2023-12-31	Schmidt, Karl (Vormieter)	P-4710
2024-01-01 – 2024-12-31	Müller, Klaus (Altmmieter)	P-4711
2025-01-01 – laufend	Weber, Anna (Neummieter)	P-4712

8.1 Ersteinrichtung — erster Mieter einzieht (NE ohne Vorgeschichte)

NE 0030 wird erstmals beim MSC eingerichtet. Nur ein Vertrag, keine Historie.

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0030",
  "partners": [
    {
      "partnerPmnumber": "P-4710",
      "name": "Schmidt",
      "firstname": "Karl",
      "commercial": "NON_COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2023-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4710"
      }
    }
  ],
  "terminateallbillingcontracts": false
}
```

8.2 Mieterwechsel — Abrechnungsempfänger (vollständige NE)

Neumieter Weber zieht am 2025-01-01 ein. Der Request enthält **alle drei Verträge** der NE — auch Schmidt und Müller, obwohl sich an deren Daten nichts geändert hat.

```

{
  "billingunitMscnumber": "123456789",
  "billingunitPmnumber": "WEG-Muster-12",
  "residentialunitMscnumber": "0010",
  "residentialunitPmnumber": "EGL-Links",
  "partners": [
    {
      "partnerPmnumber": "P-4710",
      "name": "Schmidt",
      "firstname": "Karl",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4711",
      "name": "Müller",
      "firstname": "Klaus",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL",
      "email": "anna.weber@example.com",
      "alternativeaddress": {
        "street": "Postfach 1234",
        "postalcode": "10115",
        "city": "Berlin",
        "country": "DEU"
      }
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2023-01-01",
      "validto": "2023-12-31",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4710"
      }
    },
    {
      "validfrom": "2024-01-01",
      "validto": "2024-12-31",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4711"
      }
    },
    {
      "validfrom": "2025-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    }
  ],
  "terminateallbillingcontracts": false
}

```

8.3 Mieterwechsel — uVI-Empfänger (separater Request, vollständige NE)

uVI-Empfänger für dieselbe NE aktualisieren. Müller erhielt 2024 Verbrauchsinfos, Weber ab 2025. Schmidt war kein uVI-Empfänger — er taucht daher weder in `consumptioninformationcontracts` noch in `partners[]` auf.

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "P-4711",
      "name": "Müller",
      "firstname": "Klaus",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL",
      "email": "anna.weber@example.com",
      "phonenumbers": [
        { "number": "+49 30 123456", "type": "MOBILE" }
      ]
    }
  ],
  "consumptioninformationcontracts": [
    {
      "validfrom": "2024-01-01",
      "validto": "2024-12-31",
      "partnerPmnumber": "P-4711"
    },
    {
      "validfrom": "2025-01-01",
      "partnerPmnumber": "P-4712"
    }
  ],
  "terminateallconsumptioninformationcontracts": false
}
```

8.4 Leerstand — Abrechnungsempfänger (vollständige NE)

Weber zieht zum 2025-05-31 aus, ab 2025-06-01 steht die Wohnung leer. Verwaltung ist Leerstand-Empfänger. Alle Vorverträge werden weiterhin mitgesendet.

```

{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "P-4710",
      "name": "Schmidt",
      "firstname": "Karl",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4711",
      "name": "Müller",
      "firstname": "Klaus",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "V-0001",
      "name": "Musterverwaltung GmbH",
      "commercial": "COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2023-01-01",
      "validto": "2023-12-31",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4710"
      }
    },
    {
      "validfrom": "2024-01-01",
      "validto": "2024-12-31",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4711"
      }
    },
    {
      "validfrom": "2025-01-01",
      "validto": "2025-05-31",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    },
    {
      "validfrom": "2025-06-01",
      "vacancy": true,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "V-0001"
      }
    }
  ]
}

```

```

    }
  ],
  "terminateallbillingcontracts": false
}

```

8.5 Zusätzlicher Empfänger und Leistungsgeber (Mietwohnung)

Mietwohnung mit zwei Mietern (Ehepartner), die beide auf der Abrechnung stehen; der Eigentümer/Vermieter ist Leistungsgeber. Ein Vertrag – der vollständige Datensatz besteht aus einem Eintrag. Postalisch versandt wird die Abrechnung nur an den Hauptempfänger **M-1001**; **M-1002** erscheint auf der Abrechnung, seine Adresse wird nicht genutzt (siehe §5.3).

```

{
  "billingunitMscnumber": "987654321",
  "residentialunitMscnumber": "0020",
  "partners": [
    {
      "partnerPmnumber": "M-1001",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "M-1002",
      "name": "Weber",
      "firstname": "Bernd",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "E-0815",
      "name": "Eigentümer GmbH",
      "commercial": "COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2023-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "M-1001"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "M-1002"
      },
      "sender": {
        "partnerPmnumber": "E-0815"
      }
    }
  ],
  "terminateallbillingcontracts": false
}

```

8.6 Alle Abrechnungsempfänger beenden (separater Request)

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [],
  "billingcontracts": [],
  "terminateallbillingcontracts": true,
  "terminateallbillingcontractsat": "2025-12-31"
}
```

8.7 Alle uVI-Empfänger beenden (separater Request)

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [],
  "consumptioninformationcontracts": [],
  "terminateallconsumptioninformationcontracts": true,
  "terminateallconsumptioninformationcontractsat": "2025-12-31"
}
```

9. Fehlerbehandlung

9.1 HTTP-Statuscodes

Status	Bedeutung	ERP-Maßnahme
200	Daten erfolgreich übernommen	<code>transactionid</code> protokollieren, Vorgang als erledigt markieren
400	Validierungsfehler (fachlich oder technisch)	<code>ValidationResponse.messages</code> auslesen, Fehler dem Sachbearbeiter anzeigen
401	Token ungültig oder abgelaufen	Token neu abrufen, Request einmalig wiederholen
403	Keine Berechtigung, die angefragten Daten mit den vorliegenden Credentials/ Berechtigungen zu liefern	Vertragsvereinbarung (inkl. Datenschutz-/AVV) mit MSC klären; keine automatische Wiederholung. Beim <code>POST</code> enthält die Antwort eine <code>ValidationResponse</code> – <code>shortmessage</code> / <code>message</code> auswerten
404	AE oder NE beim MSC unbekannt	MSC-Nummern aus BUD-Setup prüfen; keine automatische Wiederholung
500	Technischer Fehler beim MSC	Exponentiellen Backoff anwenden; MSC mit <code>transactionid</code> (falls vorhanden) kontaktieren

Status	Bedeutung	ERP-Maßnahme
501	GET nicht unterstützt	Kein Fehler — GET-Funktionalität im ERP deaktivieren / ignorieren

9.2 ValidationResponse — shortmessage Routing

shortmessage	Typische Ursache bei OSR	Lösungsansatz
MISSING	<code>partnerPmnumber</code> in Vertrag referenziert, aber nicht in <code>partners[]</code>	Alle referenzierten Partner vollständig in <code>partners[]</code> aufnehmen
INCOMPLETE	Pflichtfeld fehlt (z.B. <code>name</code> , <code>commercial</code> , <code>validfrom</code> , <code>owner</code>)	Datenqualität im ERP prüfen
NOT_EXPECTED	Rollentyp für diese AE nicht vereinbart (z.B. uVI nicht freigeschaltet)	MSC kontaktieren; bis zur Aktivierung nur <code>billingcontracts</code> senden
NOT_EXPECTED	Gleichzeitige Übermittlung von <code>billingcontracts</code> und <code>consumptioninformationcontracts</code> nicht erlaubt	Je Dienstleistungstyp getrennten Request senden
IMPLAUSIBLE	Datumskonflikte (<code>validto</code> vor <code>validfrom</code>), ungültiger Ländercode, ungültige IBAN	Daten korrigieren

`message` ist MSC-spezifisch und nicht standardisiert. `shortmessage` ist der verlässliche Routing-Key für programmgesteuerte Fehlerbehandlung im ERP.

9.3 Retry-Strategie

Netzwerkfehler / Timeout (keine HTTP-Antwort):

- Sofortiger Retry mit identischem Body (POST ist idempotent)
- Nach 3 Fehlversuchen: in Retry-Queue, erneuter Versuch nach 5 Min

HTTP 500:

- Exponentieller Backoff: 1 Min, 5 Min, 15 Min, 1 Std
- Nach 4 Versuchen: manuelle Eskalation + MSC kontaktieren

HTTP 400 (Validierungsfehler):

- KEIN automatischer Retry
- Fehlermeldung im ERP anzeigen, manueller Korrekturworkflow

HTTP 401:

- Token erneuern, einmaliger Retry
- Kein weiterer Retry bei erneutem 401

HTTP 403 / 404:

- KEIN automatischer Retry
- Alarm auslösen, manuelle Klärung erforderlich

9.4 Häufige Fehler und ihre Ursachen

Symptom	Wahrscheinliche Ursache	Lösung
400 MISSING bei jedem POST	Partner in Vertrag referenziert, aber nicht im <code>partners[]</code> -Array	Mapping prüfen: alle <code>partnerPmnumber</code> aus BC/CIC müssen in <code>partners[]</code> sein
Aktive Rollen verschwinden nach POST	POST enthielt nicht alle laufenden Verträge — nur Teilmenge gesendet	Immer vollständigen aktuellen Datensatz der NE senden
403 trotz gültigem Token	Vertragsvereinbarung (inkl. Datenschutz-/AVV) fehlt für diese AE oder Rollentyp	MSC kontaktieren
400 NOT_EXPECTED für CIC	<code>consumptioninformationcontracts</code> für diese AE nicht aktiviert	Nur <code>billingcontracts</code> senden bis MSC aktiviert hat
400 IMPLAUSIBLE bei Mieterwechsel	<code>validto</code> eines alten Vertrags überlappt mit <code>validfrom</code> des neuen	Datumslogik im ERP prüfen — Lückenlosigkeit und Überschneidungsfreiheit
Länderkürzel DE abgelehnt	Muss ISO 3166-1 alpha-3 sein: DEU statt DE	Ländercode-Mapping im ERP anpassen
IBAN mit Leerzeichen abgelehnt	IBAN muss ohne Leerzeichen/Trennzeichen gesendet werden	Leerzeichen vor dem Senden entfernen

10. Logging-Empfehlungen

Folgende Daten je OSR-Aufruf in einer audit-sicheren Log-Datenstruktur speichern:

Feld	Pflicht	Zweck
Zeitstempel (ISO 8601)	Ja	Reihenfolge bei Mieterwechseln rekonstruieren
<code>billingunitMscnumber</code>	Ja	Nachvollziehbarkeit, MSC-Support
<code>residentialunitMscnumber</code>	Ja	Nachvollziehbarkeit
HTTP-Methode	Ja	GET vs. POST unterscheiden
HTTP-Status	Ja	Fehleranalyse
<code>transactionid</code> aus Response	Ja (bei 200)	MSC-Support-Anfragen identifizieren
Anzahl <code>partners</code> gesendet	Empfohlen	Vollständigkeitsprüfung
Anzahl <code>billingcontracts</code> gesendet	Empfohlen	Vollständigkeitsprüfung
<code>terminateallbillingcontracts</code> Wert	Empfohlen	Audit bei Komplettkündigungen
Fehlermeldungen bei 400	Ja	Fehlerverfolgung
Benutzer / Trigger-Quelle	Empfohlen	Wer/was hat den Request ausgelöst

11. Implementierungs-Checkliste

Setup

- ☐ OAuth 2.0 Client Credentials vom MSC erhalten
- ☐ `tokenUrl` des MSC konfiguriert
- ☐ Basis-URL des MSC konfiguriert
- ☐ MSC-Nummern (`billingunitMscnumber`, `residentialunitMscnumber`) für alle NE aus BUD-API verfügbar
- ☐ Vertragsvereinbarung (inkl. Datenschutz-/AVV) mit MSC abgeschlossen

Datenmapping

- ☐ ERP-Personenstamm → `OnSitePartner` gemappt (Pflichtfelder: `partnerPmnumber`, `name`, `commercial`)
- ☐ `commercial` -Kennzeichen aus ERP auf `COMMERCIAL` / `NON_COMMERCIAL` / `UNKNOWN` abgebildet
- ☐ Abrechnungsempfänger → `BillingContract` gemappt (`validfrom`, `validto`, `vacancy`, `billingrecipient`)
- ☐ Leerstand explizit als `vacancy: true` mit Empfänger gemappt
- ☐ Ländercodes auf ISO 3166-1 alpha-3 umgestellt (z.B. `DEU` statt `DE`)
- ☐ IBAN-Formatierung: Leerzeichen entfernen vor dem Senden
- ☐ Telefonnummern auf zulässige Zeichen geprüft (nur Ziffern, `+`/`()`/`-` und Leerzeichen)
- ☐ uVI-Empfänger → `ConsumptionInformationContract` gemappt (falls relevant)

Konsistenzprüfungen (vor dem Senden)

- ☐ Alle `partnerPmnumber` aus Verträgen sind in `partners[]` enthalten
- ☐ Keine doppelte `partnerPmnumber` innerhalb `partners[]`
- ☐ Kein `terminateall*: true` gleichzeitig mit befülltem Vertrags-Array
- ☐ Datumskonsistenz: `validto` ≥ `validfrom` in jedem Vertrag
- ☐ Keine Überschneidungen in `billingcontracts` (lückenlos, überschneidungsfrei)

Trigger-Logik

- ☐ Einzug → OSR-POST ausgelöst
- ☐ Auszug / Mieterwechsel → OSR-POST ausgelöst
- ☐ Leerstand → OSR-POST ausgelöst
- ☐ Adressänderung → OSR-POST ausgelöst
- ☐ Namensänderung eines Partners → OSR-POST ausgelöst
- ☐ Ersteinrichtung (pro NE nach BUD-Setup) → OSR-POST ausgelöst

Fehlerbehandlung

- ☐ 200 → transactionid wird protokolliert
- ☐ 400 → ValidationResponse ausgelesen, shortmessage ausgewertet, Sachbearbeiter-Anzeige implementiert
- ☐ 401 → Token-Refresh + einmaliger Retry implementiert
- ☐ 403 / 404 → Alarm ohne automatischen Retry
- ☐ 500 → Exponentieller Backoff implementiert
- ☐ 501 (GET) → wird als Normal-Zustand behandelt (kein Fehler)
- ☐ Netzwerkfehler → idempotenter Retry implementiert

Tests

- ☐ POST Billing – neuer Mieter (Pflichtfelder) → 200 erhalten
- ☐ POST CIC – neuer uVI-Empfänger (separater Request) → 200 erhalten
- ☐ POST Billing – Mieterwechsel (zwei aufeinanderfolgende BC) → 200 erhalten
- ☐ POST CIC – Mieterwechsel uVI-Empfänger (separater Request) → 200 erhalten
- ☐ POST Billing – Leerstand → 200 erhalten
- ☐ POST mit BC und CIC beide befüllt → 400 erhalten (korrekte Ablehnung)
- ☐ Fehlender Partner → 400 MISSING erhalten und korrekt verarbeitet
- ☐ Datumsfehler → 400 IMPLAUSIBLE erhalten und korrekt verarbeitet
- ☐ Token-Ablauf → Token-Refresh und Retry erfolgreich
- ☐ terminateallbillingcontracts: true mit leerer Liste → 200 erhalten
- ☐ terminateallbillingcontracts: true mit gefüllter Liste → 400 erhalten
- ☐ terminateallconsumptioninformationcontracts: true (separater Request) → 200 erhalten
- ☐ GET 501 → wird ohne Fehler verarbeitet

12. Testempfehlungen

Testumgebung

Für Integrationstests steht die **bved Testplattform** zur Verfügung. Sie bildet das Verhalten eines echten MSC nach und antwortet auf OSR-Anfragen mit realistischen Mock-Daten.

- **Testplattform / Swagger UI:** <https://bved-api.de/testplatform/swagger-ui/index.html>

Empfohlene Testszenarien

Test	Erwartetes Ergebnis
POST Billing – neuer Mieter (Pflichtfelder, kein CIC)	200 + transactionid
POST CIC – uVI-Empfänger setzen (separater Request, kein BC)	200 + transactionid
POST Billing – Mieterwechsel (2 BC, 2 Partner, kein CIC)	200 + transactionid
POST CIC – Mieterwechsel uVI (separater Request, kein BC)	200 + transactionid
POST Billing – Leerstand	200 + transactionid
POST mit BC und CIC beide befüllt	400 (korrekte Ablehnung)
POST ohne partners[] (aber BC mit Referenz)	400 + MISSING
POST terminateallbillingcontracts + nicht-leere BC-Liste	400
POST validto < validfrom	400 + IMPLAUSIBLE
POST terminateallbillingcontracts (BC leer, separater Request)	200 + transactionid
POST terminateallconsumptioninformationcontracts (CIC leer, separater Request)	200 + transactionid
GET (wenn MSC unterstützt)	200 mit den beim MSC gespeicherten Daten
GET (wenn MSC nicht unterstützt)	501 , kein Applikationsfehler

Anhang A: Referenz zu Version 3.10

Dieser Anhang ordnet die Abrechnungsempfänger-Konstellationen des abgelösten **Standard-Datenaustauschs V 3.10** den Feldern der On-Site Roles API v2.0 zu. Er richtet sich an ERP-Anbieter, die eine bestehende 3.10-Anbindung auf die REST-APIs umstellen und die vertrauten Fälle wiederfinden wollen.

Fundstelle: Standard-Datenaustausch V 3.10 (dritte Erweiterung, September 2025), Abschnitt 5, **Satzart M = Standardnutzersatz, Feld 7 „Kennzeichen Adressfeld“** (Länge 1, Position 52, Pflichtfeld, Seite 17) — „Kennzeichen zum Andrucken der Adresse auf der Einzelabrechnung“.

A.1 Mapping-Tabelle

Kennz.	Bedeutung in V 3.10	Abbildung in OSR v2.0	Andruck
1	Nutzeranschrift	Nutzer als <code>billingrecipient (owner: false)</code> , keine <code>alternativeaddress</code>	Name des Nutzers, Anschrift der Nutzeinheit
2	Eigentümeranschrift	Eigentümer als <code>billingrecipient (owner: true)</code> mit <code>alternativeaddress</code>	Name und Anschrift des Eigentümers
3	Eigentümername / Nutzername / Eigentümeranschrift	Eigentümer als <code>billingrecipient (owner: true)</code> , Nutzer als <code>additionalbillingrecipient (owner: false)</code>	Name und Anschrift des Eigentümers, Nutzername als Zusatzzeile
4	Leistungsnehmer	—	—

Zu Kennzeichen 4: Das Kennzeichen war in V 3.10 definiert, wurde in der Praxis aber nie genutzt. Es ist in dieser Zuordnung bewusst ausgeklammert und hat keine Entsprechung in der OSR API.

Adressierung: Nur der `billingrecipient` ist der postalische Empfänger — bei gesetzter `alternativeaddress` dessen abweichende Postadresse, andernfalls die Adresse der Nutzeinheit. Der `additionalbillingrecipient` wird lediglich auf der Abrechnung genannt; seine Adresse wird nicht verwendet (siehe §5.3).

Alle Beispiele beziehen sich auf dieselbe Nutzeinheit (`123456789` / `0010`) und verwenden die Partner aus §8, ergänzt um zwei Eigentümer und einen weiteren Nutzer.

A.2 Kennzeichen 1 — Nutzeranschrift

Fachlicher Fall: Mieter erhält die Abrechnung.

Abbildung: Nutzer als `billingrecipient ( owner: false )`, keine `alternativeaddress`.

Andruck: Adressfeld Anna Weber, Anschrift der Nutzeinheit.

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2025-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    }
  ],
  "terminateallbillingcontracts": false
}
```

A.3 Kennzeichen 2 — Eigentümeranschrift

Fachlicher Fall: Eigentümer erhält die Abrechnung.

Abbildung: Eigentümer als `billingrecipient` (`owner: true`) mit `alternativeaddress`; kein zusätzlicher Empfänger — der Nutzer erscheint nicht auf der Abrechnung.

Andruck: Adressfeld Eigentümer GmbH, Hauptstr. 1, 80331 München.

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "E-0815",
      "name": "Eigentümer GmbH",
      "commercial": "COMMERCIAL",
      "alternativeaddress": {
        "street": "Hauptstr. 1",
        "postalcode": "80331",
        "city": "München",
        "country": "DEU"
      }
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2025-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0815"
      }
    }
  ],
  "terminateallbillingcontracts": false
}
```

A.4 Kennzeichen 3 – Eigentümername / Nutzernamen / Eigentümeranschrift

Fachlicher Fall: Eigentümer erhält die Abrechnung, der Nutzer wird auf der Abrechnung genannt.

Abbildung: Eigentümer als `billingrecipient` (`owner: true`) mit `alternativeaddress`, Nutzer als `additionalbillingrecipient` (`owner: false`) – dessen Adresse wird nicht verwendet.

Andruck: Adressfeld Eigentümer GmbH, Hauptstr. 1, 80331 München; Zusatzzeile Nutzer: Anna Weber.

```
{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "E-0815",
      "name": "Eigentümer GmbH",
      "commercial": "COMMERCIAL",
      "alternativeaddress": {
        "street": "Hauptstr. 1",
        "postalcode": "80331",
        "city": "München",
        "country": "DEU"
      }
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2025-01-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0815"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    }
  ],
  "terminateallbillingcontracts": false
}
```

A.5 Kennzeichen 3 mit Nutzerwechsel

Fachlicher Fall: Der Eigentümer bleibt, der Nutzer wechselt. Abgrenzung zum 30.06.

Abbildung: Zwei lückenlose, überschneidungsfreie Zeitscheiben mit demselben `billingrecipient` und wechselndem `additionalbillingrecipient`.

Andruck: Zwei Abrechnungen an Eigentümer GmbH, Hauptstr. 1, 80331 München; mit Zusatzzeile Anna Weber bzw. Jürgen Hoffmann.

```

{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "E-0815",
      "name": "Eigentümer GmbH",
      "commercial": "COMMERCIAL",
      "alternativeaddress": {
        "street": "Hauptstr. 1",
        "postalcode": "80331",
        "city": "München",
        "country": "DEU"
      }
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    },
    {
      "partnerPmnumber": "P-4713",
      "name": "Hoffmann",
      "firstname": "Jürgen",
      "commercial": "NON_COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2025-01-01",
      "validto": "2025-06-30",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0815"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    },
    {
      "validfrom": "2025-07-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0815"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4713"
      }
    }
  ],
  "terminateallbillingcontracts": false
}

```

A.6 Kennzeichen 3 mit Eigentümerwechsel

Fachlicher Fall: Der Eigentümer wechselt, der Nutzer bleibt. Der Nutzungszeitraum von Anna Weber läuft durch; die Zeitscheiben bestimmen, an welchen Eigentümer die jeweilige Abrechnung geht.

Abbildung: Zwei lückenlose, überschneidungsfreie Zeitscheiben mit wechselndem `billingrecipient` und demselben `additionalbillingrecipient`.

Andruck: Zwei Abrechnungen — bis 30.06. an Eigentümer GmbH, Hauptstr. 1, 80331 München; ab 01.07. an Petra Becker, Lindenweg 5, 81675 München; jeweils mit Zusatzzeile Nutzer: Anna Weber.

```

{
  "billingunitMscnumber": "123456789",
  "residentialunitMscnumber": "0010",
  "partners": [
    {
      "partnerPmnumber": "E-0815",
      "name": "Eigentümer GmbH",
      "commercial": "COMMERCIAL",
      "alternativeaddress": {
        "street": "Hauptstr. 1",
        "postalcode": "80331",
        "city": "München",
        "country": "DEU"
      }
    },
    {
      "partnerPmnumber": "E-0816",
      "name": "Becker",
      "firstname": "Petra",
      "commercial": "NON_COMMERCIAL",
      "alternativeaddress": {
        "street": "Lindenweg 5",
        "postalcode": "81675",
        "city": "München",
        "country": "DEU"
      }
    },
    {
      "partnerPmnumber": "P-4712",
      "name": "Weber",
      "firstname": "Anna",
      "commercial": "NON_COMMERCIAL"
    }
  ],
  "billingcontracts": [
    {
      "validfrom": "2025-01-01",
      "validto": "2025-06-30",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0815"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    },
    {
      "validfrom": "2025-07-01",
      "vacancy": false,
      "billingrecipient": {
        "owner": true,
        "partnerPmnumber": "E-0816"
      },
      "additionalbillingrecipient": {
        "owner": false,
        "partnerPmnumber": "P-4712"
      }
    }
  ]
}

```

```
"terminateallbillingcontracts": false  
}
```