



Wärmenetzportal



MANUAL

Wärmenetzportal

Benutzer-, Integrations- und Betriebshandbuch für Superadmins,
Firmen-Admins, Kunden und Analytics-Admins



Inhaltsverzeichnis

1	Zu diesem Manual	3
1.1	Zielgruppen	3
1.2	Verbindliche Begriffe	3
2	Produktüberblick	3
2.1	Drei Oberflächen, ein fachlicher Ablauf	3
2.2	Warum PostgreSQL und DuckDB getrennt bleiben	3
2.3	Rollen- und Datengrenzen	4
3	Anmeldung und Orientierung	4
3.1	Portaladressen	4
3.2	Navigation	4
3.3	Sitzungen und Passwort-Reset	5
4	Superadmin: Mandanten in Betrieb nehmen	5
4.1	Mandant anlegen	5
4.2	Analytics provisionieren	5
4.3	Abnahmecheck	6
5	Firmen-Admin: Kunden und Zähler	6
5.1	Kundenstamm	6
5.2	Verträge und physische Zähler	6
5.3	CSV-Import	6
5.4	Analytics-Zählerzuordnung	6
6	Firmen-Admin: Jahresverbrauch und Rechnungen	6
6.1	Gemeinsame Jahrestabelle	6
6.2	Tarife	7
6.3	Abschläge	7
6.4	Einzelrechnung erzeugen	7
6.5	Unveränderliche Rechnungsbasis	8
6.6	Dokumente und Anfragen	8
7	Kundenportal	8
7.1	Anmeldung und Dashboard	8
7.2	Berechnungsregeln	9
7.3	Rechnungen und Dokumente	9
8	Analytics	9
8.1	Einstieg per SSO	9
8.2	Sensoren und Schnellansicht	10
8.3	Pipeline für einen Zeitreihenplot	10
8.4	Abrechnungsprojektion	11
8.5	Abrechnungstichtage prüfen	11
9	Uploader installieren und betreiben	11
9.1	Lieferartefakt und Voraussetzungen	11
9.2	Minimale Konfiguration	11
9.3	Sichere Erstinbetriebnahme	11
9.4	Outbox- und Crash-Vertrag	12
9.5	Zählerhistorie	12
9.6	Historischer Backfill	12
10	Ingest-API	12
10.1	Portal-Endpunkt	12
10.2	Kumulativer Stand	12
10.3	Periodenverbrauch ohne Rohdaten	12
10.4	HTTP-Antworten	13
11	Sicherheit und Datenschutz	13
11.1	Technische Kontrollen	13
11.2	Datenminimierung	13
11.3	Bewusste Restgrenzen	13
12	Backup, Deployment und Wiederherstellung	13
12.1	Getrennte Restore-Einheiten	13
12.2	Vor dem Deploy	14
12.3	Nach dem Deploy	14
13	Fehlerdiagnose	14
13.1	Rechnung kann nicht erzeugt werden	14
13.2	Analytics-Pipeline öffnet nicht	14
13.3	Uploader sendet nicht	14
13.4	Browser meldet CSP-Fehler mit fremdem utils.js	14
14	Abnahmeprotokoll	15
14.1	Kontakt	15

I 1 Zu diesem Manual

Dieses Manual beschreibt den dokumentierten Produktstand des Wärmenetzportals, der gekoppelten Analytics-Seite und des lokalen Uploaders. Es ist zugleich Anwenderhandbuch, Integrationsleitfaden und Betriebsrunbook. Mandantenspezifische Verträge, Preise, Aufbewahrungsfristen und Verantwortlichkeiten haben Vorrang vor allgemeinen Beispielen.

1.1 Zielgruppen

Rolle	Aufgabe	Relevante Kapitel
Superadmin	Mandanten, Benutzerkonten, Branding, Betrieb	2, 4, 11–14
Firmen-Admin	Kunden, Zähler, Tarife, Abrechnung, Exporte	5–8
Analytics-Admin	Zeitreihen, Pipelines, Nutzer und API-Keys	9–10
Kunde	Eigene Verbräuche, Rechnungen und Dokumente	8
Technischer Betreiber	Uploader, Deploy, Backup, Monitoring	10–14

1.2 Verbindliche Begriffe

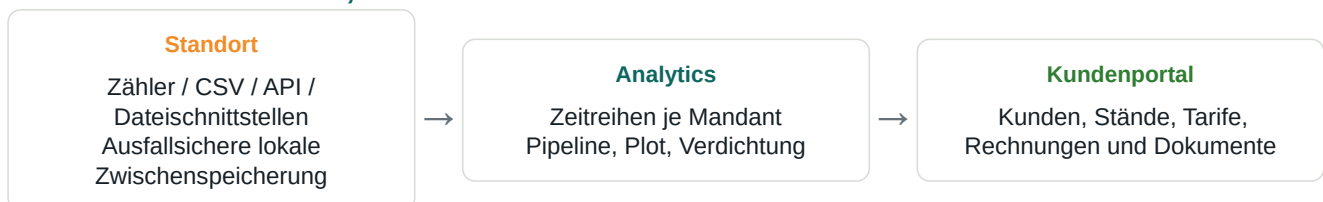
- **Mandant / Tenant:** ein organisatorisch und technisch getrennter Wärmenetzbetreiber.
- **Portal:** PostgreSQL-basierte Anwendung für Identitäten, Kunden, abrechnungsfähige Werte, Tarife, Rechnungen und Dokumente.
- **Analytics:** tenantbezogene Anwendung unter `analytics.waermenetzportal.de` mit Sensorserien, Pipelines und Plot-Ausgaben.
- **Uploader:** eigenständiges Standortprogramm aus dem Lieferpaket; es ist kein Pipeline-Modul.
- **Freigegebener Wert:** `accepted` und abrechnungsrelevant. `estimated` bleibt Entwurf und ist weder Kundenausgabe noch Rechnungsgrundlage.

Keine implizite Schätzung.

Das Portal interpoliert fehlende Tages- oder Stichtagswerte nicht. Eine Rechnung wird nur aus einer eindeutigen, freigegebenen und für den kompletten Zeitraum passenden Grundlage erzeugt.

I 2 Produktüberblick

2.1 Drei Oberflächen, ein fachlicher Ablauf



Am Standort liest der Uploader Werte und Zählerhistorien, schreibt sie zuerst dauerhaft in seine lokale SQLite-Outbox und sendet offene Batches synchron an Analytics. Dort bleiben hochaufgelöste Rohzeitreihen optional erhalten. Eine kontrollierte Pipeline erzeugt Tagesverbräuche oder exakte Periodenverbräuche für das Portal. Das Portal liefert Kundenansicht, Jahresübersicht und Abrechnung.

2.2 Warum PostgreSQL und DuckDB getrennt bleiben

PostgreSQL ist für konkurrierende Portalzugriffe, relationale Integrität, Kundenbeziehungen, Nummernkreise und Dokumentmetadaten zuständig. DuckDB ist für große analytische Zeitreihen und reproduzierbare Pipelines je Tenant geeignet. Dass beide Dienste auf demselben Server laufen, ändert ihre fachliche Datenhoheit nicht. Analytics serialisiert schreibende Datenbankabschnitte in einem Dienstprozess durch einen globalen Mutex. HTTP-Uploads bleiben synchron: Erfolg wird erst nach dem Commit bestätigt. Der Uploader muss deshalb lokal wiederholbar sein; seine Outbox ist die dauerhafte Client-Queue. Eine zweite Serverqueue und ein zweistufiges HTTP-Protokoll sind nicht erforderlich.

2.3 Rollen- und Datengrenzen

Portal-Superadmin

Globale Portalrolle; Mandanten anlegen und verwalten.
Kein globaler Zugriff auf Analysedaten.

Firmen-Admin

Nur eigener Tenant; Kunden, Zähler, Rechnungen und SSO zu Analytics.

Analytics-Admin

Nur ein aktiver Analytics-Mandant; Nutzer, API-Keys und Pipelines im Tenant.

Kunde

Portalzugriff nur auf eigene `customer_id`; kein Analytics-Zugang.

3 Anmeldung und Orientierung

3.1 Portaladressen

Zentrale Seite	https://waermenetzportal.de
Firmenlogin	<a href="https://waermenetzportal.de/<firmen-slug>">https://waermenetzportal.de/<firmen-slug>
Analytics	https://analytics.waermenetzportal.de
Superadmin	https://waermenetzportal.de/admin/login

Die Firmenadresse ist bereits die Loginseite. Ein Firmen-Admin und ein Kunde können dieselbe Seite verwenden; die serverseitig hinterlegte Rolle bestimmt das Ziel nach dem Login. Alte `/<slug>/login`-Links werden dauerhaft auf die kanonische Mandantenseite umgeleitet.

Die sichtbare Anmeldung ist zweistufig. Zuerst wird die **E-Mail-Adresse oder Kundennummer** eingetragen und **Einloggen** gewählt; danach erscheint das Passwortfeld. Ein zweiter Klick auf **Einloggen** meldet an. Firmen-Admins, Superadmins und Kunden verwenden denselben Ablauf. Für Kunden ist die E-Mail-Adresse im Kundenstamm maßgeblich; eine getrennte Rechnungs-E-Mail wird nicht geführt.

Neukunde steht neben **Einloggen** und öffnet auf derselben öffentlichen Mandantenseite ein Formular für Name, Verbrauchsstelle, E-Mail/Telefon, gewünschten Wechseltermin und Nachricht. Die Anfrage landet in **Dokumente & Anfragen** des Firmen-Admins. Honeypot, Kurzsperrung und ein cookieunabhängiges Mandant-/IP-Limit begrenzen Missbrauch. **Anfrage senden** und **Zurück** stehen nebeneinander. Die Betreiberlinks unter der Box sind zentriert und durch Mittelpunkte getrennt.

3.2 Navigation

Der Firmen-Admin sieht **Status**, **Kunden**, **Dokumente**, **Analysen**, **Einstellungen** und **Anleitung**. Jahresverbrauch und Rechnungsstatus sind unter **Kunden** zusammengeführt; Rechnungslauf und Freigabe befinden sich unter **Dokumente**. Der aktuell geöffnete Navigationspunkt ist farbig markiert und für Hilfsttechnologien mit `aria-current` ausgezeichnet. Mit **Abmelden** am Ende der linken Navigation wird die Sitzung beendet; dieselbe Aktion ist auch in der globalen Superadmin-Oberfläche vorhanden.

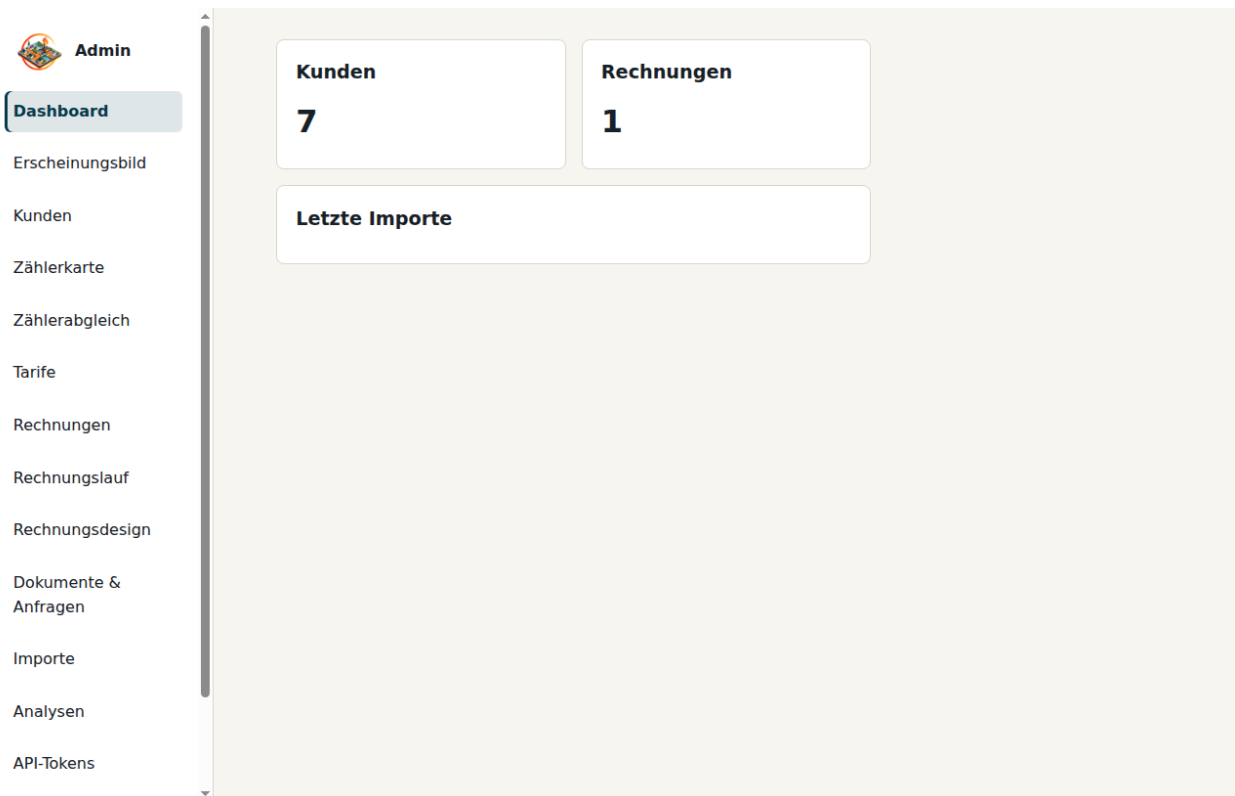


Figure 1: Firmen-Admin mit synthetischen Demodaten.

3.3 Sitzungen und Passwort-Reset

Session-Cookies sind `HttpOnly`, `SameSite=Lax` und unter `HTTPS` `Secure`. Loginfehlversuche werden serverseitig und nicht nur im Browser begrenzt. Der Reset-Endpunkt speichert gehashte Einmaltoken, versendet aber noch keine Reset-Mail. Benachrichtigungen über veröffentlichte Rechnungen/Dokumente und Kundenanfragen laufen getrennt über eine transaktionale Mail-Outbox. Kundenkonten müssen weiterhin über einen kontrollierten Provisionierungsweg angelegt werden.

4 Superadmin: Mandanten in Betrieb nehmen

4.1 Mandant anlegen

- 1 **Anmelden.** `/admin/login` mit dem persönlichen Plattformkonto öffnen.
- 2 **Mandant erfassen.** Name, kanonischen Slug, Status und Brandingfarben eintragen. Slugs nach Produktivstart nicht beiläufig ändern.
- 3 **Betreiberlogo setzen.** Als Firmen-Admin **Erscheinungsbild** öffnen und ein PNG-, JPEG- oder WEBP-Logo hochladen. Es erscheint im Login und im Neukundenformular; das Rechnungslogo bleibt eine getrennte Einstellung.
- 4 **Rechtslinks prüfen.** Impressum und Datenschutz des Betreibers müssen auf dessen Loginseite anonym erreichbar sein.
- 5 **Firmen-Admin verbinden.** Ein persönliches Admin-Konto exakt diesem Tenant zuordnen; keine geteilten Alltagskonten verwenden.
- 6 **Login abnehmen.** Mandantenseite samt Betreiberlogo, Firmen-Admin-Ziel und abgewiesenen Fremdzugriff testen.

4.2 Analytics provisionieren

Der Firmen-Admin öffnet im Portal **Analysen**. Das Portal sendet serverseitig ein kurzlebiges HMAC-Einmalticket an Analytics; der Browser erhält danach die tenantgebundene Analytics-Session. Es gibt kein

zweites Portalpasswort. Der Portal-Superadmin wird in Analytics nur als Admin des gerade ausgewählten Tenants provisioniert, nicht als globaler Datenbank-Superadmin.

4.3 Abnahmecheck

- Mandant ist aktiv und besitzt eindeutigen Slug.
- Betreiberlogo gehört zum richtigen Mandanten; ohne Upload greift das gemeinsame Fallback.
- Firmen-Admin sieht ausschließlich diesen Mandanten.
- Kunde landet im Kundenportal, Firmen-Admin im Adminbereich.
- **Analysen** öffnet genau den richtigen Analytics-Mandanten.
- **Anleitung** zeigt Superadmins, Betreibern und Kunden jeweils nur die für ihre Rolle bestimmten Abläufe.
- Vor Import oder Migration liegt eine geprüfte Restore-Einheit vor.

5 Firmen-Admin: Kunden und Zähler

5.1 Kundenstamm

Unter **Kunden** kann der Admin suchen, neu anlegen und bearbeiten. Relevante Felder sind Kundennummer, Name, Adresse, physische Zählernummer, Verbrauchsstelle, Abschlagsbetrag und Ansprechpartner.

Zählernummern sind innerhalb eines Tenants eindeutige physische Identitäten. Eine Änderung im Kundenformular synchronisiert Kundenzeile und kanonische Zählertabelle transaktional. Ein bereits einem anderen Kunden zugeordneter Zähler wird abgelehnt. Kundenwechsel erfordern belegte Vertrags- und Übergabezeiträume; aus einem Messwertplateau darf kein Datum geraten werden.

5.2 Verträge und physische Zähler

Am Kunden werden Tarif, Vertragsbeginn, optionales Vertragsende, physische Zähler, Startmonat des Abrechnungsjahres und die tatsächliche Zahl der Abschläge gepflegt. Januar–Dezember verwendet Startmonat 1, August–Juli Startmonat 8. Leer bei der Abschlagszahl bedeutet Tarifautomatik.

Verträge desselben Kunden und Zuordnungen desselben physischen Zählers dürfen sich nicht überlappen. Derselbe Zähler kann aber nacheinander Alt- und Neueigentümer bedienen. Rechnung und Kundenansicht verwenden dann nur den belegten Vertragsabschnitt, während die physische Messreihe erhalten bleibt. Eine endgültige Schlussrechnung ist pro Vertrag und Abrechnungszyklus eindeutig und wird von einem späteren Jahreslauf übersprungen.

5.3 CSV-Import

Der Button **Importieren** unter **Kunden** übernimmt ausschließlich Kundenstammdaten. Vor dem Speichern zeigt das Portal eine Vorschau mit allen Konflikten. Bestehende Kundennummern oder Zähler werden nie überschrieben; beim Bestätigen werden nur konfliktfreie Zeilen importiert. Verbrauchs- und Rechnungsdateien lassen sich in der Oberfläche nicht hochladen.

Datenquelle vor Geschwindigkeit.

Namen, Adressen, Kundennummern, Zählernummern, Verbräuche, Abschläge und Vertragszeiträume sind unterschiedliche fachliche Quellen. Ein Verbrauchs-PDF ist keine zulässige Quelle für Abschläge oder Messwerte, wenn dies nicht ausdrücklich belegt ist.

5.4 Analytics-Zählerzuordnung

Analytics darf statt der Portal-Zählernummer seine stabile externe Sensoridentität senden. Das Portal löst sie nur über eine aktive tenantgebundene `meter_sources`-Zuordnung auf. Unbekannte externe Identitäten legen weder Kunden noch Zähler an. Das verhindert, dass Quellbezeichnungen zur zweiten Zählerwahrheit werden.

6 Firmen-Admin: Jahresverbrauch und Rechnungen

6.1 Gemeinsame Jahrestabelle

Unter **Rechnungen** wählt der Admin ein Abrechnungsjahr. Die Tabelle wird in ihrer vollständigen intrinsischen Breite gezeichnet; weder Tabelle noch ihr direkter Wrapper scrollen selbst. Horizontal scrollt die umgebende Admin-Hauptspalte. Links stehen Kundennummer, Name, Adresse, Zähler, Verbrauchsstelle, Anschlussleistung, Anfangs-/Endstand und Jahresverbrauch. Rechts folgen Rechnungsnummer, Rechnungsdatum, Betrag und **PDF** oder **erzeugen**. Die frühere Spalte **Strang** wird nicht angezeigt.

The screenshot displays the 'Kunden, Zähler, Tarif oder Rechnung durchsuchen' search results table and the 'Rechnungen' form. The search results table lists customer data, including customer number, meter number, and consumption data. The 'Rechnungen' form shows fields for customer selection, time period, and calculation date, along with a preview of the resulting bill.

Figure 2: Kunden- und Verbrauchsübersicht sowie Rechnungserstellung mit Dokumentvorschau.

Der Verbrauchsexport liefert CSV, Excel oder PDF. Er enthält ausschließlich die fachlichen Verbrauchszeilen, ohne Gesamtverbrauch, Strangsummen oder Rechnungsmetadaten. Der Rechnungslauf unter **Dokumente** ist der Bulk-Einstieg für mehrere Kunden und frei wählbare Zeiträume.

6.2 Tarife

Unterstützt werden:

- Nettopreis je kWh plus monatlicher Grundpreis.
- Brutttopreis je MWh plus Grundpreis je kW Anschlussleistung.
- Nettopreis je MWh plus Jahresgrundpreis je kW Anschlussleistung. Der Jahresgrundpreis wird taggenau und centstabil auf Vertragsabschnitte verteilt.
- Umsatzsteuer und konfigurierbare Zahl der Abschlagsperioden.
- Zeitliche Gültigkeit `valid_from` / `valid_to`.

Die gedeckelten Tarife verwenden in diesem dritten Modell 48,18 EUR/MWh netto und 21,01 EUR/kW/Jahr netto für Altkunden beziehungsweise 33,61 EUR/kW/Jahr netto für Neukunden. Bei 14 kW sind das 294,14 EUR oder 470,54 EUR Jahresgrundpreis netto. Gültigkeit, Steuersatz und Vertragszuordnung sind separate freizugebende Fachdaten.

Ein explizit gewählter Tarif muss aktiv und am Abrechnungsende gültig sein. **Automatisch** wählt nur einen passenden Tarif. Ohne passende Grundlage nennt die Fehlermeldung den konkreten Stichtag.

6.3 Abschläge

Wenn der Tarif Abschlagsperioden vorsieht, muss im Kundenstamm ein Abschlagsbetrag gepflegt sein. NULL bedeutet **unbekannt** und blockiert die Rechnung. Ein ausdrücklich gepflegter Wert 0,00 EUR ist dagegen eine vollständige fachliche Angabe. Abschläge dürfen nicht aus Anschlussleistung, Tarif oder einer fremden Referenzrechnung geschätzt werden.

Die am Vertrag eingetragene Zahl überschreibt die Tarifautomatik. Der Wert 0 bedeutet ausdrücklich, dass im Vertragsabschnitt keine Abschläge vereinnahmt wurden; dafür ist kein Abschlagsbetrag erforderlich. Ein leeres Feld übernimmt die Zahl aus dem Tarif.

6.4 Einzelrechnung erzeugen

- 1 **Jahr prüfen.** In **Rechnungen** den passenden Jahreszeitraum auswählen und beide Grenzstände plausibilisieren.
- 2 **Kunde öffnen.** In derselben Zeile **erzeugen** klicken. Das Formular ist auf diesen Tenant und Kunden festgelegt.
- 3 **Nummer und Parameter prüfen.** Die nächste Rechnungsnummer, Zeitraum, Rechnungsdatum, Tarif und Vorlage kontrollieren. Das Präfix stammt aus dem Rechnungsdesign; die laufende Nummer gehört zum Rechnungslauf.
- 4 **Erzeugen.** Die konkrete Backendmeldung vollständig lesen; ein rotes generisches „Fehler“ ohne Ursache ist kein erwarteter Zustand.
- 5 **PDF abnehmen.** Stände, Verbrauch, Preisbestandteile, Steuer, Abschläge, Forderung und Nummer prüfen.

6

Veröffentlichen. In **Rechnungen** die intern geprüfte PDF markieren und **Ausgewählte veröffentlichen** klicken. Erst jetzt erscheint sie beim Kunden und die Benachrichtigung wird vorgemerkt.

7

Satz laden. Mit **Alle PDFs als ZIP** den endgültigen Rechnungssatz des gewählten Jahres herunterladen.

Eine Startnummer, die mit demselben Präfix bereits vergeben wurde oder unter der höchsten vergebenen Nummer liegt, wird abgelehnt. Gleichzeitige Läufe desselben Mandanten werden in PostgreSQL serialisiert. Jede erfolgreiche Kundenrechnung reserviert ihre Nummer trotzdem in ihrer eigenen Transaktion; ein fachlich gescheiterter Kunde verbraucht keine Nummer.

6.5 Unveränderliche Rechnungsbasis

Bei Erfolg speichert die Rechnung einen JSON-Snapshot der verwendeten Abrechnungsbasis und eine Berechnungsversion. Spätere Messwert- oder Tarifänderungen verändern die vorhandene PDF nicht rückwirkend. Rechnungen und Dokumente liegen außerhalb des Webroots; der Downloadendpunkt prüft Rolle, Tenant und bei Kunden zusätzlich `customer_id`.

6.6 Dokumente und Anfragen

Unter **Dokumente & Anfragen** lädt der Admin ein PDF entweder für alle aktiven Kunden oder genau einen Kunden hoch. Das Dokument erscheint im Kundenportal; für erreichbare Empfänger wird eine deduplizierte Benachrichtigung vorgemerkt. Im selben Posteingang werden Neukunden-, Kontakt-, Eigentümer-/Mieterwechsel- und Störungsmeldungen mit den Zuständen offen, in Bearbeitung, erledigt oder abgelehnt geführt.

7 Kundenportal

7.1 Anmeldung und Dashboard

Der Kunde öffnet dieselbe Mandantenseite wie der Firmen-Admin und trägt E-Mail oder Kundennummer sowie sein persönliches Passwort ein. Nach dem Login lädt das Dashboard alle benötigten Daten mit einem einzigen GET `/api/<slug>/overview`. Es findet kein zweiter Login und kein separater Request je Diagramm statt.

Der Kopf zeigt vollen Namen und Kundennummer in einer Zeile. Links im globalen Seitenkopf stehen Betreiberlogo und Betreibername, rechts beendet **Abmelden** die Sitzung. Die Oberfläche ist in **Übersicht**, **Dokumente**, **Daten**, **Eigentümer-/Mieterwechsel** und **Störungsmeldung** gegliedert. Die Übersicht nennt Bezugsdatum beziehungsweise -monat direkt in jeder Kennzahl und zeigt den Monatsverbrauch als Diagramm. Unter **Dokumente** stehen Verträge/Dokumente links und veröffentlichte Abrechnungen rechts. Im Tab **Daten** werden die aktuell gespeicherte E-Mail-Adresse und Telefonnummer sichtbar angezeigt und können im selben Formular geändert werden. Änderungen sowie Anträge benachrichtigen den Admin.

Sichtbar sind ausschließlich freigegebene eigene Daten:

- kumulative Zählerstände,
- exakt abgrenzbare Tages-, Monats- und Jahresverbräuche,
- explizite freigegebene Periodenwerte,
- Datenlücken mit Zahl fehlender Tage,
- eigene Rechnungen und sichtbare Dokumente.

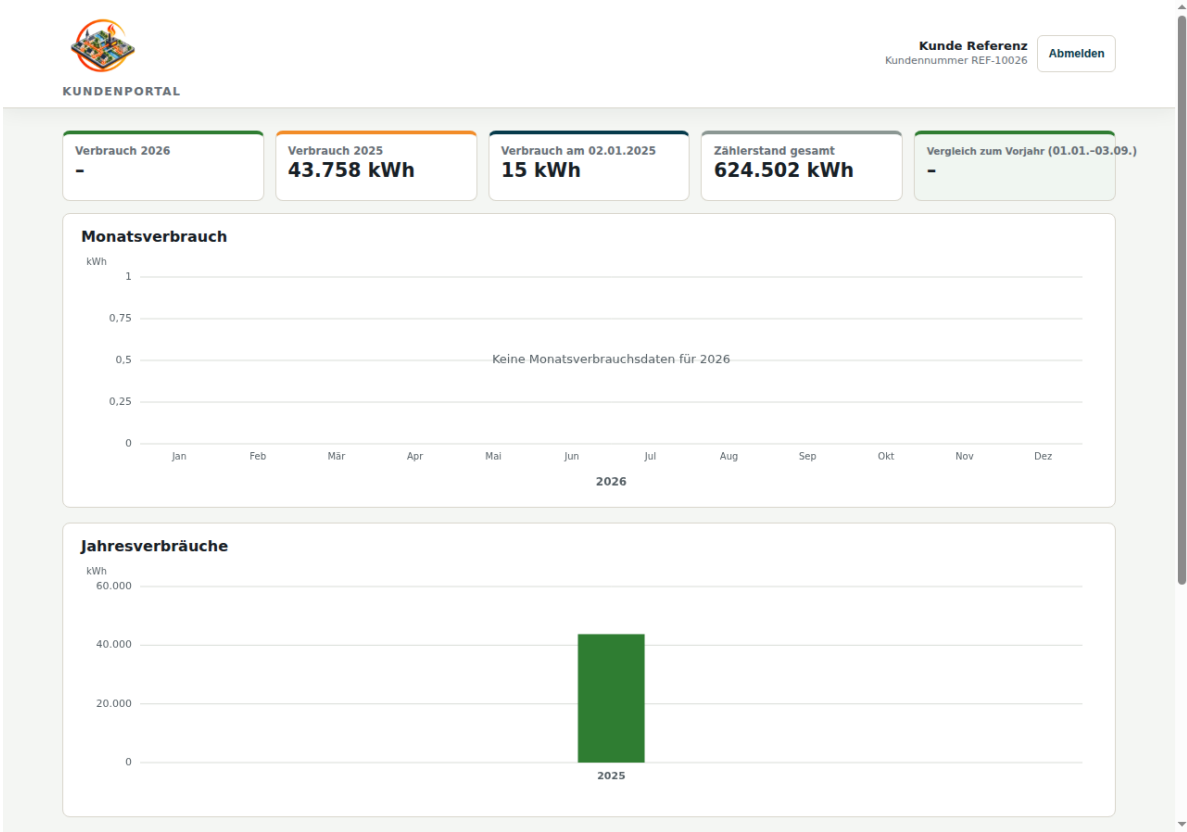


Figure 3: Synthetisches Kunden-Dashboard mit Verbrauch und Ständen.

7.2 Berechnungsregeln

Anzeige	Regel
Tag	Differenz zweier akzeptierter direkt aufeinanderfolgender Kalendertagsgrenzen desselben Zählers.
Monat	Differenz aus exaktem letzten Tag des Vormonats und exaktem letzten Tag des Monats.
Jahr	Differenz aus exaktem 31.12. des Vorjahres und exaktem 31.12. des Jahres.
Periodenwert	Ein akzeptierter exakt passender expliziter Wert ist autoritativ.
Rücklauf	Die Rechnung korrigiert die Endstandsdifferenz nur mit den in PostgreSQL markierten Rücklauftagen; sie benötigt keinen DuckDB-Zugriff.
Lücke	Fehlende Grenzen oder eine unvollständige Rücklaufprojektion erzeugen keine erfundene Verbrauchszeile und blockieren die Rechnung.

7.3 Rechnungen und Dokumente

Im Rechnungsbereich kann der Kunde nur eigene PDFs öffnen. Eine fremde oder nicht zugeordnete ID liefert 404, damit keine Existenzinformation über andere Kunden offengelegt wird. Dokumente können kundenspezifisch oder tenantweit freigegeben sein.

8 Analytics

8.1 Einstieg per SSO

Firmen-Admins melden sich unter `waermenetzportal.de/<anbieter>` an und öffnen dort **Analysen**. Ein direkter Passwort-Login unter der Analytics-Domain ist in der gehosteten Umgebung deaktiviert. Portal-Endkunden werden nie in Analytics provisioniert.

8.2 Sensoren und Schnellansicht

Unter **Daten** werden Sensorgruppen und Serien gewählt. Die Schnellansicht ist für eine begrenzte visuelle Prüfung gedacht; große Zeiträume gehören in eine Pipeline mit expliziter Auswahl, Ausrichtung und Plot-Ausgabe.

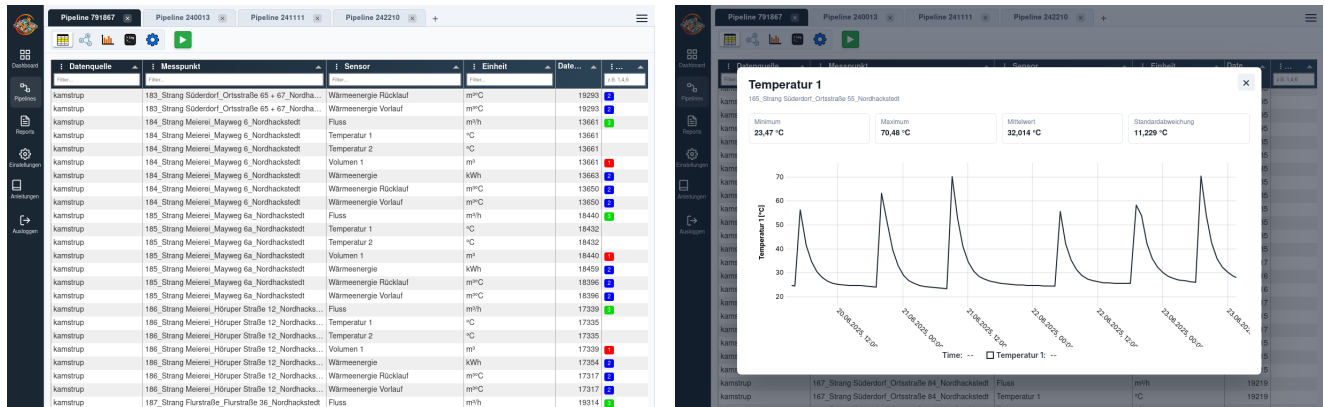


Figure 4: Sensorauswahl und Schnellansicht in Analytics.

8.3 Pipeline für einen Zeitreihenplot

- 1 **Daten einlesen.** Tenant, Sensoren und Zeitraum wählen.
- 2 **Align.** Zeitachsen und optionales Raster fachlich festlegen.
- 3 **Plot Time Series.** Serien, Achsen und Beschriftung konfigurieren.
- 4 **Display.** Ergebnis im Browser anzeigen und Plausibilität prüfen.
- 5 **Speichern.** Eindeutigen Namen, Zweck, Zeitraumlogik und Eigentümer dokumentieren.

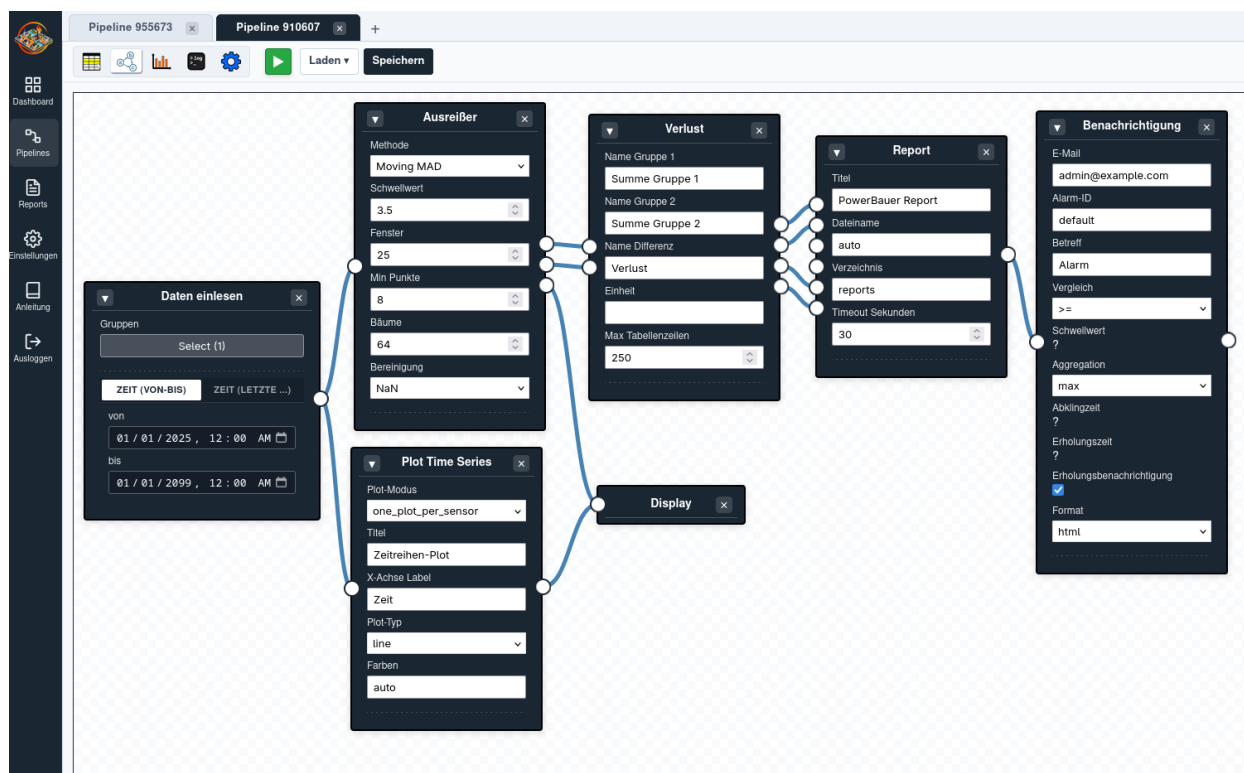


Figure 5: Analytics-Pipeline: Daten einlesen, verarbeiten, ausgeben und speichern.

8.4 Abrechnungsprojektion

Das admin-only Modul `portal_billing` verdichtet ausgewählte kumulative Serien zu Tagesverbräuchen oder einem exakten Periodenverbrauch. Portalziel und Token kommen nur aus der Serverumgebung; eine Pipeline kann kein beliebiges Ziel oder Secret einschleusen. Der tägliche Sollplan läuft um 02:00 Europe/Berlin mit einem rollierenden 35-Tage-Fenster und schließt den laufenden Tag aus. Verspätete Providerwerte werden über stabile Quellreferenzen idempotent nachgezogen. Eine Tageslücke wird je Zähler gemeldet und hält die übrigen Zähler nicht zurück. Vor einer Rechnung ist trotzdem der vollständige Zeitraum erneut abzugleichen.

8.5 Abrechnungsstichtage prüfen

Das admin-only Modul **Abrechnungsstichtage prüfen** wird ohne vorgeschalteten Datenknoten angelegt. Der Admin wählt die kumulativen Zählergruppen, echten Zeitraum, Europe/Berlin und die Toleranz (Standard $\pm 3,25$ Stunden). Für Januar–Dezember prüft es 01.01. 00:00 an beiden Grenzen; für August–Juli 01.08. 00:00. Es liest ausschließlich diese beiden Toleranzfenster.

Der Textausgang wird mit **E-Mail** im Modus **nur wenn Text nicht leer** verbunden. Fehlende Zähler erscheinen mit Zählernummer, Sensorname/Adresse und fehlender Grenze; bei vollständigen Daten bleibt der Ausgang leer und es wird nichts versandt. Abweichende Abrechnungsjahre erhalten getrennte Pipelines.

9 Uploader installieren und betreiben

9.1 Lieferartefakt und Voraussetzungen

Es existiert genau ein kanonisches Uploader-Programm für Linux und eine aus demselben Quellstand gebaute Windows-Ausgabe. Anbieter werden über passende Datenquellen angebunden, ohne den Cloud-Datenfluss auf einzelne Hersteller festzulegen. Der Uploader benötigt ausgehendes HTTPS, eine geschützte Konfiguration, einen eigenen Analytics-API-Key und ein dauerhaftes Verzeichnis für die SQLite-Outbox.

9.2 Minimale Konfiguration

```
{
  "server": "https://analytics.waermenetzportal.de",
  "tenant": "<tenant-slug>",
  "api_key": "pbk_<einmaliger-schlüssel>",
  "outbox_file": "uploader_outbox.sqlite3",
  "batch_size": 5000,
  "interval_seconds": 900,
  "readouts": [
    { "type": "<konfigurierte-datenquelle>", "start": "2025-01-01T00:00:00Z" }
  ]
}
```

Feldnamen und Readout-Details werden mit `--setup` erzeugt; das Beispiel ist kein Ersatz für den interaktiven Setup-Dialog. Konfiguration und Outbox sind personen- und verbrauchsbezogene Betriebsdaten und mindestens mit Modus `0600` zu schützen.

9.3 Sichere Erstinbetriebnahme

Der konkrete Dateiname steht im Lieferpaket; `uploader` bezeichnet in den folgenden Befehlen das ausgelieferte Programm.

```
uploader --list-readouts
uploader --setup --config uploader.json
chmod 600 uploader.json
uploader --config uploader.json --test-connection
uploader --config uploader.json --dry-run
uploader --config uploader.json --server-dry-run
uploader --config uploader.json --once
uploader --config uploader.json --status
```

Modus	Semantik
<code>--test-connection</code>	Erreichbarkeit und Key mit leerem Punktearray; schreibt nichts.
<code>--dry-run</code>	Liest Providerdaten und berichtet lokal; persistiert keine Outbox-Änderung.
<code>--server-dry-run</code>	Holt frische Werte und sendet sie zur Servervalidierung; weder Server-DB noch lokale Outbox werden verändert.
<code>--once</code>	Ein echter Capture-/Persist-/Sendezyklus.

--daemon	Wiederholt den normalen Zyklus im konfigurierten Intervall.
--status	Exakte read-only Outbox-Zähler; Uploader vorher stoppen.

9.4 Outbox- und Crash-Vertrag

Neue Messwerte, Capture-Cursor, Jobs und Zählerhistorien werden in einer BEGIN IMMEDIATE-Transaktion atomar in SQLite geschrieben. Erst nach diesem Commit beginnt die Netzübertragung. Jeder Punkt bleibt pending, bis genau seine HTTP-Bestätigung vorliegt. Ein Prozessabbruch kann deshalb zu einer idempotenten Wiederholung, aber nicht zu einem still verlorenen Punkt führen.

Bestätigte Werte bleiben 365 **Messtage** lokal. Pending-, Konflikt- und Fehlerzeilen werden nicht altersbedingt gelöscht. SQLite läuft mit WAL und synchronous=FULL. Vor Dateikopie, --status oder einer read-only SQLite-Inspektion muss der schreibende Uploader gestoppt werden. Outbox, -wal und -shm bilden gemeinsam den Zustand. Zwei Uploaderprozesse dürfen nie dieselbe Outbox öffnen; mehrere Uploader desselben Tenants mit jeweils eigener Outbox sind zulässig.

9.5 Zählerhistorie

Soweit eine Datenquelle eine Zählerhistorie liefert, verarbeitet der Uploader alle Einträge statt nur des ersten. Zeitlich begrenzte Sensoridentitäten und Metadaten werden versioniert in derselben SQLite-Outbox gespeichert. Beim Upgrade einer alten Outbox wird vorhandene Historie aus bereits gespeicherten Identitäten übernommen. Die Outbox darf bei einem Upgrade nicht gelöscht oder durch eine leere Datei ersetzt werden. Zugangstoken werden nicht in der Outbox gespeichert.

9.6 Historischer Backfill

Ein Backfill verwendet eine eigene Konfigurationsdatei und eine eigene outbox_file; er ersetzt nie den normalen Betriebszustand. Ablauf:

1. normalen Uploader stoppen und Zustand sichern;
2. separate Backfill-Konfiguration mit benötigtem UTC-Start erstellen;
3. --dry-run, danach --server-dry-run ausführen;
4. genau einen beaufsichtigten --once-Lauf starten;
5. Status und Serverbestand prüfen; normale Outbox nicht zusammenführen.

I 10 Ingest-API

10.1 Portal-Endpunkt

```
POST /api/<tenant-slug>/ingest/consumption
Authorization: Bearer <prefix>.<secret>
Content-Type: application/json
```

Der Admin erzeugt unter **API-Tokens** ein Token mit Scope consumption:write. Das Secret erscheint nur einmal und wird serverseitig nur als Hash gespeichert. Ein Batch enthält höchstens 5.000 Einträge. Unbekannte, inaktive oder fremdmandantliche Zähler werden abgelehnt und niemals automatisch angelegt.

10.2 Kumulativer Stand

```
[
  {
    "kind": "reading",
    "meter_number": "M-1001",
    "reading_date": "2026-12-31",
    "value": 1410.5,
    "unit": "MWh",
    "quality": "measured",
    "source_system": "operator",
    "source_reference": "annual-M-1001-2026"
  }
]
```

10.3 Periodenverbrauch ohne Rohdaten

```
[
  {
    "kind": "period_consumption",
    "meter_number": "M-2001",
    "period_start": "2026-01-01",
```

```
"period_end": "2026-12-31",  
"value": 43.758,  
"unit": "MWh",  
"quality": "derived",  
"source_system": "operator",  
"source_reference": "annual-2026-M-2001-v2",  
"derivation_version": "operator-export-v2"  
}  
]
```

Damit können Betreiber die Rohzeitreihe bei sich behalten und nur eine freigegebene abgeleitete Abrechnungsgröße senden. Mehrere gleichzeitig akzeptierte Quellen für denselben Zähler und Zeitraum bleiben nachvollziehbar, blockieren aber die Rechnung bis zur fachlichen Auflösung.

10.4 HTTP-Antworten

- 200: alle Zeilen akzeptiert; Client prüft zusätzlich `ok=true` und `failed=0`.
- 207: Teilfehler; erfolgreiche Zeilen bleiben gespeichert, fehlerhafte werden mit stabilen Referenzen gezielt wiederholt.
- 401/403: Key, Scope, Tenant, Ablauf oder HTTPS prüfen.
- 409: abweichender Wert für bereits belegte Identität/Zeit fachlich klären.
- 413: Batch verkleinern, ohne offene Nachbarzeilen aus der Outbox zu löschen.

11 Sicherheit und Datenschutz

11.1 Technische Kontrollen

- Prepared Statements für alle SQL-Zugriffe.
- `tenant_id` an mandantenspezifischen Datensätzen.
- `tenant_id` + `customer_id` für Kundenendpunkte.
- CSRF-Token für schreibende Browserrequests.
- Loginlimit in PostgreSQL, unabhängig von neuen Cookies.
- Gehashte API- und Reset-Token.
- PDF-Signatur- und Bilddekodierungsprüfung bei Uploads.
- Private Dokumente außerhalb des Webroots.
- Restriktive CSP; keine anwendungseigenen Inline-Skripte.
- SSO-Einmaltickets mit kurzer Lebensdauer und Einmalverwendung.
- Interne Rechnungen sind bis zur expliziten Veröffentlichung für Kunden unsichtbar; Veröffentlichung und Mail-Outbox sind eine Transaktion.
- SMTP läuft ohne Shell und ohne Zugangsdaten in Prozessargumenten; Produktion verlangt verschlüsseltes SMTP.

11.2 Datenminimierung

Rohzeitreihen sind im Portal nicht erforderlich. Betreiber können nur abgeleitete Periodenwerte übertragen. Logs dürfen keine Passwörter, API-Keys, OAuth-Token, SSO-Tickets oder vollständigen fremden Response-Body enthalten. Die lokale Outbox enthält Verbrauchs- und Zählerhistoriendaten und ist wie eine Fachdatenbank zu behandeln.

11.3 Bewusste Restgrenzen

- Passwort-Reset/Provisionierung versendet noch keine Reset-Mail; Rechnungs-, Dokument- und Anfragebenachrichtigungen sind davon getrennt vorhanden.
- Externe/off-site Sicherung und Secret-Rotation sind Betriebsaufgaben.
- Ein gemischter Ingest-Batch ist zeilenweise, nicht gesamtatomar.
- Serienrechnungen bleiben gesperrt, solange autoritative Fachdaten fehlen.
- Last- und Verfügbarkeitszusagen werden projektbezogen vereinbart und sind nicht aus Referenzbestandszahlen abzuleiten.

12 Backup, Deployment und Wiederherstellung

12.1 Getrennte Restore-Einheiten

System	Zu sichernder Zustand
Portal	PostgreSQL-Custom-Dump, <code>pg_restore -l</code> -Inhaltsverzeichnis und private Rechnungs-/Dokumentdateien in einem SHA-256-Manifest.

Analytics	Kontrolldaten und mandantenspezifische Analysedaten in projektlokalen Snapshot.
Uploader	Gestoppte Konfiguration, SQLite-Outbox und vorhandene WAL/SHM-Dateien als gemeinsame Einheit.

Ein Portal-Deploy erstellt vor Migrationen einen verifizierten projektlokalen PostgreSQL-Custom-Dump, führt danach idempotente Migrationen aus und darf bestehende Fachdaten oder private Dateien nicht überschreiben. Ein Analytics-Deploy ersetzt Anwendungsartefakte, aber nicht mandantenspezifische Daten, Uploaderkonfiguration oder Outbox. Kein Deploy richtet automatisch ein externes Backup ein.

12.2 Vor dem Deploy

1. Arbeitsbaum und Commit identifizieren.
2. Relevante Syntax-, Unit-, PostgreSQL-, Browser- und SSO-Tests ausführen.
3. Projektlokale Restore-Einheit erzeugen und Manifest prüfen.
4. Sicherstellen, dass keine unvollständige Bulk-Abrechnung läuft.
5. Erst danach Code synchronisieren, Migrationen ausführen und Dienste neu starten.

12.3 Nach dem Deploy

- /healthz sowie Portal- und Analytics-Dienststatus prüfen.
- Login als Firmen-Admin und Kunde testen.
- Jahrestabelle horizontal scrollen und PDF geschützt öffnen.
- Portal → Analytics-SSO und Beispielpipeline testen.
- uploader --status bei gestopptem Uploader prüfen.
- Backupmanifest, installierte Commit-/Dateihashes und Logs dokumentieren.

13 Fehlerdiagnose

13.1 Rechnung kann nicht erzeugt werden

Meldung	Prüfung
Kein gültiger Tarif	Abrechnungsende gegen valid_from, valid_to und Aktivstatus prüfen.
Stand zu weit vom Stichtag	Zeitraum und exakte Grenzwerte prüfen; keinen beliebigen letzten Wert verwenden.
Abrechnungsstände fehlen	Zähleridentität, Zählerhistorie, Lieferzeitraum der Datenquelle und Freigabestatus prüfen.
Abschlagsbetrag fehlt	Autoritativen Kundenwert pflegen; NULL nicht mit geschätztem Nullwert ersetzen.
Konkurrierende Periodenwerte	Quellen fachlich auflösen; keine Zeile still löschen.

13.2 Analytics-Pipeline öffnet nicht

- Browsercache hart neu laden und installierten Commit prüfen.
- /api/modules und Mandantensession prüfen.
- Gespeicherte Node-Typen gegen verfügbare Module vergleichen.
- Das Template-Registry-Objekt muss vor der Moduldiscovery als leere Map existieren; ein leerer Modulrequest darf keine JavaScript-Ausnahme auslösen.

13.3 Uploader sendet nicht

- Dienst stoppen und --status read-only ausführen.
- DNS, TLS-Uhrzeit, Tenant und Key mit --test-connection prüfen.
- pending, conflict und failed getrennt betrachten.
- Aktives -wal niemals unter laufendem Prozess ignorieren.
- Bei 207 nur die gemeldeten Fehler fachlich bearbeiten; erfolgreiche Serverzeilen bleiben gespeichert.
- Outbox nie löschen, um „noch einmal von vorne“ zu beginnen.

13.4 Browser meldet CSP-Fehler mit fremdem utils.js

Das Portal verwendet script-src 'self' und keine eigenen Inline-Skripte. Eine Meldung mit zufälliger UUID und fremdem utils.js kann von einer Browsererweiterung stammen. Gegenprobe ist ein sauberes privates Profil ohne Add-ons. Fremde Hashes werden nicht in die Portal-CSP aufgenommen.

I 14 Abnahmeprotokoll

Nr.	Prüfung	Status
1	Tenant, Rollen und Loginziele	☐
2	Kunden-/Zählerimport mit Dry-run	☐
3	Uploader Test → Dry-run → Server-Dry-run → Once	☐
4	365-Tage-Outbox und Zählerhistorie nach Neustart	☐
5	Analytics-SSO und gespeicherte Plot-Pipeline	☐
6	Tages-, Monats-, Jahreswerte und sichtbare Lücken	☐
7	Referenzrechnung vollständig nachgerechnet	☐
8	Kundenisolation und Fremddownload 404	☐
9	Portal-/Analytics-/Uploader-Restoreeinheiten	☐
10	Betriebsverantwortliche und Eskalationsweg dokumentiert	☐

14.1 Kontakt



Rabenaber GmbH
Am Steinig 11, 74939 Zuzenhausen
info@rabener.com
<https://www.rabenaber.com>

Copyright © 2026 Rabenaber GmbH. Dieses Manual beschreibt den dokumentierten Softwarestand und ersetzt weder individuelle Versorgungsverträge noch fachliche, rechtliche oder steuerliche Freigaben für eine konkrete Abrechnung.