

Planung: Budget, Arbeitspakete

- [Projektumfang und Arbeitspakete 15.01.](#)
- [Projektumfang und Arbeitspakete 17.02](#)
- [IST Stand 14.01.](#)
- [Abnahmetest auf Entwicklungssystem für Arbeitspaket #1](#)
- [Project and Release Planung:](#)
- [Workshop - Next steps](#)

Projektumfang und Arbeitspakete 15.01.

Gesamtüberblick

Projektumfang

1. Datenimport aus Gridware

- Automatisierter Import von Gridware-Daten in ERPNext per CSV
- Automatisches Abrufen der monatlichen CSV-Dateien und Import in ERPNext
- Aufbereitung, Validierung und Mapping der Daten auf die entsprechenden ERPNext DocTypes

1.1 Verbrauchsdaten

- Import und Befüllen von Verbrauchsdaten (Consumption Data Entries)
- Strukturierte Ablage je Kunde und Abrechnungszeitraum

1.2 Kundenstammdaten

- Importieren und Anlegen von Neukunden

1.3 Kundenstammdaten

- Importieren, Anlegen und Verlinken von:
 - Kontaktdaten
 - Adressdaten

1.3 SEPA-Mandate & Bankdaten

- Importieren, Anlegen und Verlinken von SEPA-Mandaten
- Anlage und Verknüpfung von Kunden-Bankkonten

2. Abonnements & Abrechnung

- Verknüpfung von Verbrauchsdaten mit Abonnements

- Nutzung der Abonnements als Grundlage für die automatisierte Rechnungserstellung

3. Rechnungserstellung

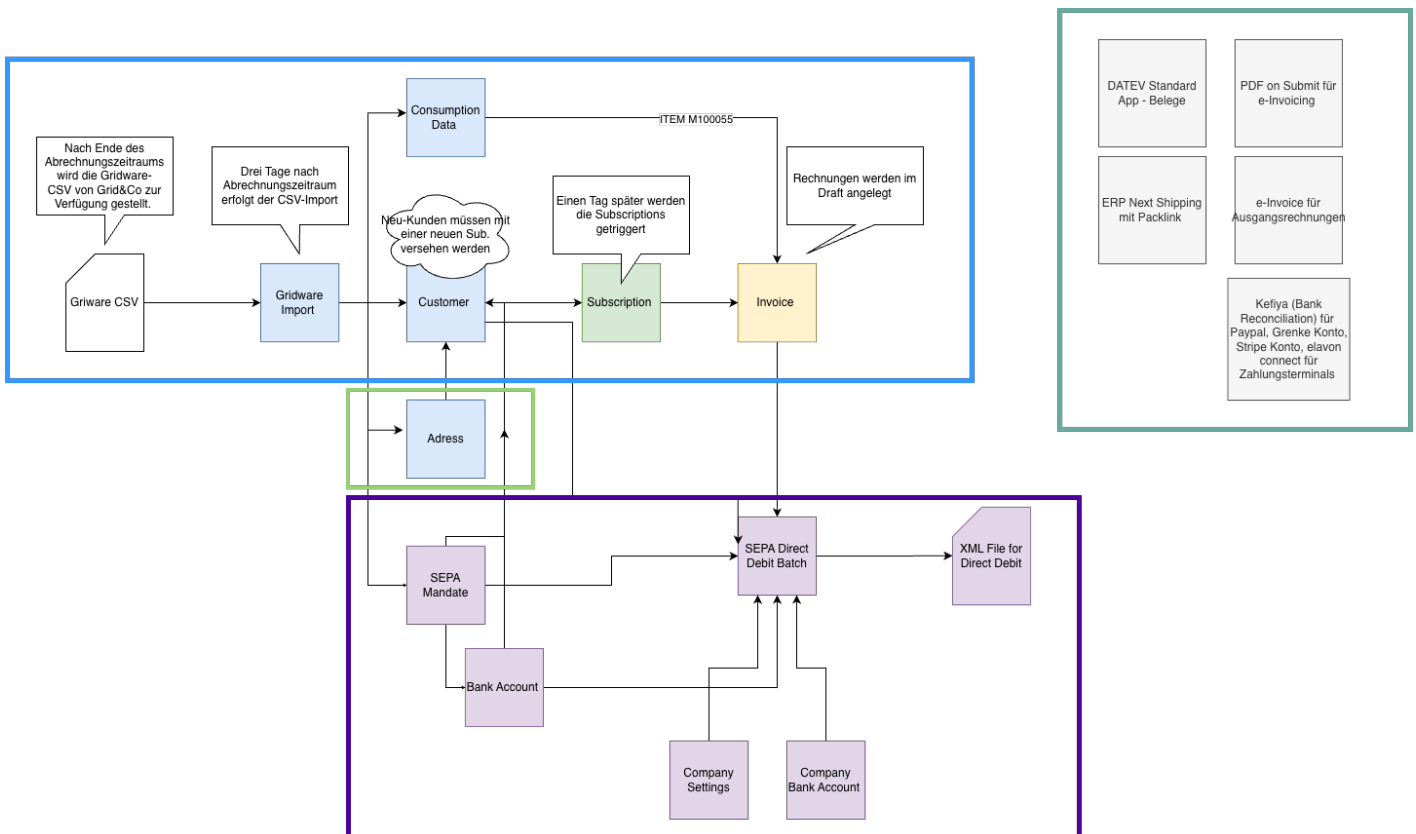
- Erstellung von Ausgangsrechnungen (Sales Invoices)
- Integration eines Verbrauchsdaten-Anhangs in die Rechnungsdokumente

4. SEPA Lastschriftverfahren

- Automatisches Befüllen des SEPA Direct Debit DocTypes aus:
 - Unternehmensdaten
 - Kundendaten
 - Rechnungs- und Transaktionsdaten
- Erstellung von SEPA XML-Dateien zur Sammelübermittlung an die Bank

5. Systemeinrichtung & Betrieb

- Installation und Konfiguration der erforderlichen Standard-Apps in ERPNext
- Systemeinrichtung und Betrieb eines lokal gehosteten Produktivsystems



Go Live Planung

Priorität 1 – Paket 1 ~10 std

Bereitstellung bis: 01.02.

Umfang

- Becharged App Funktionalität
 - Datenimport aus Gridware CSV Datei
 - Importieren und Befüllen von Verbrauchsdaten
 - Importieren und Anlegen von Neukunden
 - Verknüpfung von Verbrauchsdaten mit Abonnements
 - Erstellung von Rechnungsdokumenten inkl. Verbrauchsdaten-Anhang
- Abnahme Test auf Entwicklungssystem
 - Ausführliches Testen mit Projektverantwortlichen (Sarah & Matthias)
- Einrichtung Produktiv-System
 - Becharged: Bereitstellung Zugriff auf Produktivsystem für Phamos, mit Möglichkeit Apps einzuspielen und Updates zu installieren (Plan via SSH)
 - Installation und Konfiguration von BeCharged Custom App
- Erstellung der Januar Abrechnung via automatisierter Verarbeitung der Gridware CSV

Priorität 2 – Paket 2 ~20 std

Installation & Konfiguration von Standard-Apps

- Zahlungsabgleich
 - Installation der Kefiya-App in ERPNext
 - Umfang: Installation & grundlegende Konfiguration
 - Anbindung der folgenden Zahlungsarten (technische Verfügbarkeit vorausgesetzt):
 - Must-haves:
 - Grenke Bank
 - PayPal (vorbehaltlich der Unterstützung durch Kefiya)
 - Nice-to-haves
 - Stripe (vorbehaltlich der Unterstützung durch Kefiya)
 - Evalon (vorbehaltlich der Unterstützung durch Kefiya)
- Finanzbuchhaltung
 - DATEV Standard App
 - Zweck: Ermöglichung der Belegübertragung an DATEV Unternehmen Online
 - Umfang: Installation der DATEV Standard App in ERPNext
 - Abgrenzung: Die Leistung beschränkt sich vorerst auf die Installation der App.
- E-Rechnung
 - e-Invoice für Ausgangsrechnungen
 - Umfang: Installation und grundlegende Konfiguration der e-Invoice-Funktionalität für Ausgangsrechnungen
 - Aufbereitung von Stammdaten: Aufbereitung und Korrektur relevanter Stammdaten durch BeCharged, um typische e-Invoice-Fehler (z. B. fehlende

- Pflichtfelder) zu vermeiden
- PDF on Submit für e-Invoicing
 - Zweck: Automatische Generierung von Rechnungs-PDFs im Rahmen des e-Invoicing-Prozesses
 - Umfang: Installation und Konfiguration der App „PDF on Submit“
- Paket-Versand
 - ERPNext Shipping mit Packlink
 - Umfang: Installation der ERPNext Shipping App (Packlink)
 - Abgrenzung: Die Leistung umfasst ausschließlich die Installation der App. Eine fachliche Einrichtung, Tarifkonfiguration oder Prozessintegration ist nicht Bestandteil dieses Pakets.
- IBAN-Prüfung und VAT-Nummern
 - Installation der App „ERPNext Germany“
 - Zweck: Bereitstellung technischer Grundlagen für:
 - IBAN-Prüfungen
 - Validierung von Umsatzsteuer-Identifikationsnummern
 - Umfang: Die Leistung beschränkt sich auf die Installation der App.

Priorität 3 – Paket 3 ~20 std

Individualentwicklung

- Automatisches Importieren der CSV Datei über Microsoft Power BI
- Datenimport aus Gridware
 - Importieren, Anlegen und Verlinken von SEPA-Mandaten und Bankkonten
- Zahlungsabwicklung via SEPA
 - Automatisches Befüllen des SEPA Direct Debit DocTypes
 - Erstellung der SEPA XML-Dateien

Priorität 4 – Paket 4 ~20 std

Individualentwicklung

- Datenimport aus Gridware
 - Importieren, Anlegen und Verlinken von Kontakt- und Adressdaten

Projektumfang und Arbeitspakete 17.02

Regeln zu Releases:

<https://becharged.phamos.eu/books/dokumentation/page/project-and-release-planung>

Gesamtüberblick

Projektumfang

☒ **1. Datenimport aus Gridware**

- Automatisierter Import von Gridware-Daten in ERPNext per CSV
- Automatisches Abrufen der monatlichen CSV-Dateien und Import in ERPNext
- Aufbereitung, Validierung und Mapping der Daten auf die entsprechenden ERPNext DocTypes

☐ **1.2 NEU Stammdaten + Verbrauchsdaten Import via API**

- Importieren von Stammdaten (Kunde, Lieferant, Kontaktdaten, Adressdaten, SEPA Mandate, Bankaccounts, Vertrag) über Swagger API
- Importieren der Transaktionsdaten über Swagger API
- Aufbereitung, Validierung und Mapping der Daten auf die entsprechenden ERPNext DocTypes
- Neue DocTypes für sauberen Datentransfer von Gridware nach ERP Next (Vertrag, Single Sessions, User)
- Lookup der Vertragskunden via Vertrag

☒ **1.1 Verbrauchsdaten**

- Import und Befüllen von Verbrauchsdaten (Consumption Data Entries)
- Strukturierte Ablage je Kunde und Abrechnungszeitraum

☒ **1.2 Kundenstammdaten**

- Importieren und Anlegen von Neukunden

☐ **1.3 Kundenstammdaten**

- Importieren, Anlegen und Verlinken von:
 - Kontaktdaten
 - Adressdaten

☐ **1.3 SEPA-Mandate & Bankdaten**

- Importieren, Anlegen und Verlinken von SEPA-Mandaten
- Anlage und Verknüpfung von Kunden-Bankkonten

☒ **2. Abonnements & Abrechnung**

- Verknüpfung von Verbrauchsdaten mit Abonnements
- Nutzung der Abonnements als Grundlage für die automatisierte Rechnungserstellung

☒ **3. Rechnungserstellung**

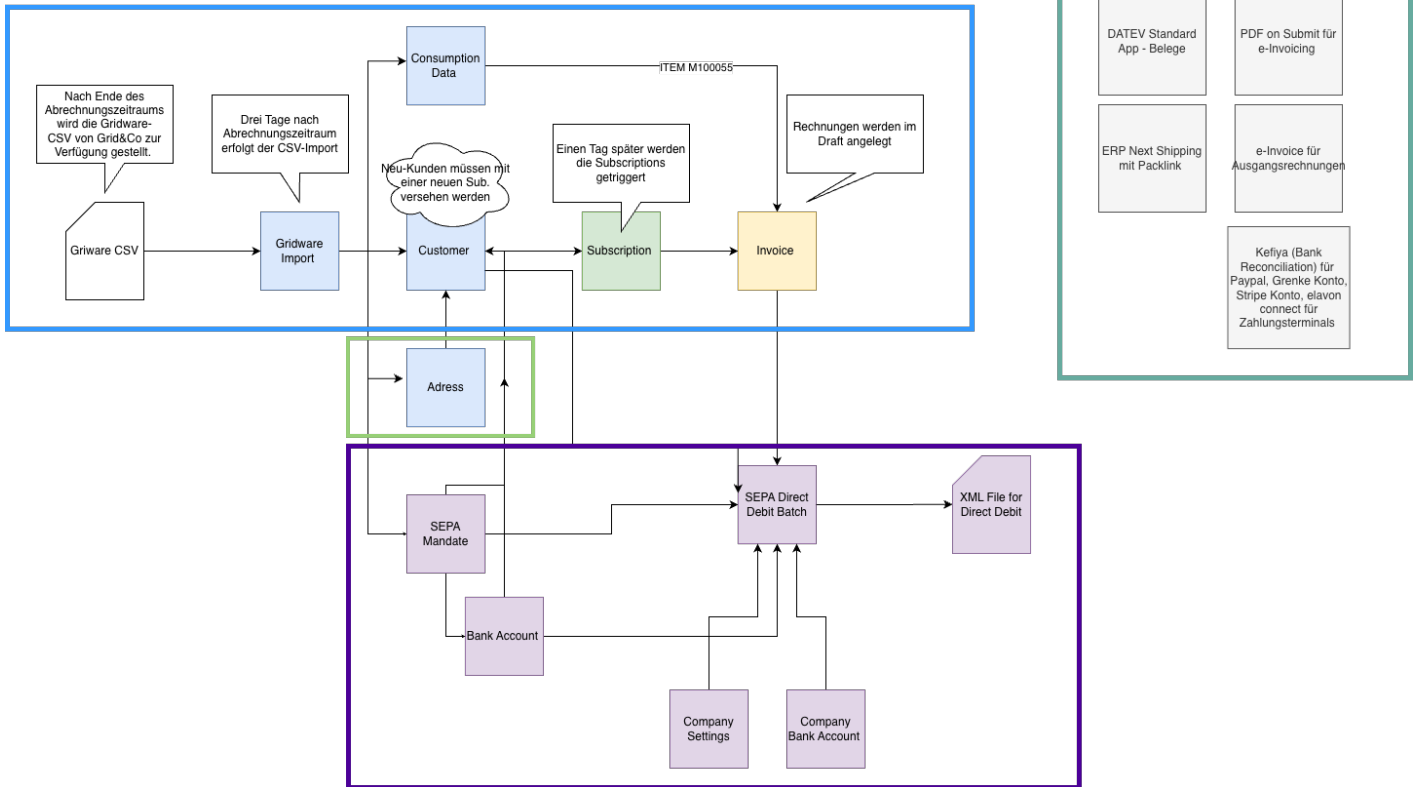
- Erstellung von Ausgangsrechnungen (Sales Invoices)
- Integration eines Verbrauchsdaten-Anhangs in die Rechnungsdokumente

☐ **4. SEPA Lastschriftverfahren**

- Automatisches Befüllen des SEPA Direct Debit DocTypes aus:
 - Unternehmensdaten
 - Kundendaten
 - Rechnungs- und Transaktionsdaten
- Erstellung von SEPA XML-Dateien zur Sammelübermittlung an die Bank

☒ **5. Systemeinrichtung & Betrieb**

- Installation und Konfiguration der erforderlichen Standard-Apps in ERPNext
- Systemeinrichtung und Betrieb eines lokal gehosteten Produktivsystems



Review: Production Release

#1

DONE Priorität 1 – Paket 1 ~10 std

Bereitstellung bis: 01.02.

Umfang

- Becharged App Funktionalität
 - Datenimport aus Gridware CSV Datei
 - Importieren und Befüllen von Verbrauchsdaten
 - Importieren und Anlegen von Neukunden
 - Verknüpfung von Verbrauchsdaten mit Abonnements
 - Erstellung von Rechnungsdokumenten inkl. Verbrauchsdaten-Anhang
- Abnahme Test auf Entwicklungssystem
 - Ausführliches Testen mit Projektverantwortlichen (Sarah & Matthias)
- Einrichtung Produktiv-System
 - Becharged: Bereitstellung Zugriff auf Produktivsystem für Phamos, mit Möglichkeit Apps einzuspielen und Updates zu installieren (Plan via SSH)

- Installation und Konfiguration von BeCharged Custom App
- Erstellung der Januar Abrechnung via automatisierter Verarbeitung der Gridware CSV

DONE Priorität 2 – Paket 2 ~20 std

Installation & Konfiguration von Standard-Apps

- Zahlungsabgleich
 - Installation der Kefiya-App in ERPNext
 - Umfang: Installation & grundlegende Konfiguration
 - Anbindung der folgenden Zahlungsarten (technische Verfügbarkeit vorausgesetzt):
 - Must-haves:
 - Grenke Bank
 - PayPal (vorbehaltlich der Unterstützung durch Kefiya)
 - Nice-to-haves
 - Stripe (vorbehaltlich der Unterstützung durch Kefiya)
 - Evalon (vorbehaltlich der Unterstützung durch Kefiya)
- Finanzbuchhaltung - **Wird von becharged aktiviert, Funktionalität ist gegeben**
 - DATEV Standard App
 - Zweck: Ermöglichung der Belegübertragung an DATEV Unternehmen Online
 - Umfang: Installation der DATEV Standard App in ERPNext
 - Abgrenzung: Die Leistung beschränkt sich vorerst auf die Installation der App.
- E-Rechnung
 - e-Invoice für Ausgangsrechnungen
 - Umfang: Installation der e-Invoice-Funktionalität für Ausgangsrechnungen
 - Aufbereitung von Stammdaten: Aufbereitung und Korrektur relevanter Stammdaten durch BeCharged, um typische e-Invoice-Fehler (z. B. fehlende Pflichtfelder) zu vermeiden
 - PDF on Submit für e-Invoicing
 - Zweck: Automatische Generierung von Rechnungs-PDFs im Rahmen des e-Invoicing-Prozesses
 - Umfang: Installation und Konfiguration der App „PDF on Submit“
- Paket-Versand
 - ERPNext Shipping mit Packlink
 - Umfang: Installation der ERPNext Shipping App (Packlink)
 - Abgrenzung: Die Leistung umfasst ausschließlich die Installation der App. Eine fachliche Einrichtung, Tarifkonfiguration oder Prozessintegration ist nicht Bestandteil dieses Pakets.
- IBAN-Prüfung und VAT-Nummern
 - Installation der App „ERPNext Germany“
 - Zweck: Bereitstellung technischer Grundlagen für:

- IBAN-Prüfungen
- Validierung von Umsatzsteuer-Identifikationsnummern
- Umfang: Die Leistung beschränkt sich auf die Installation der App.

70% DONE Priorität 3 – Paket 3 ~20 std

Individualentwicklung

On Prod. bis 27.02.

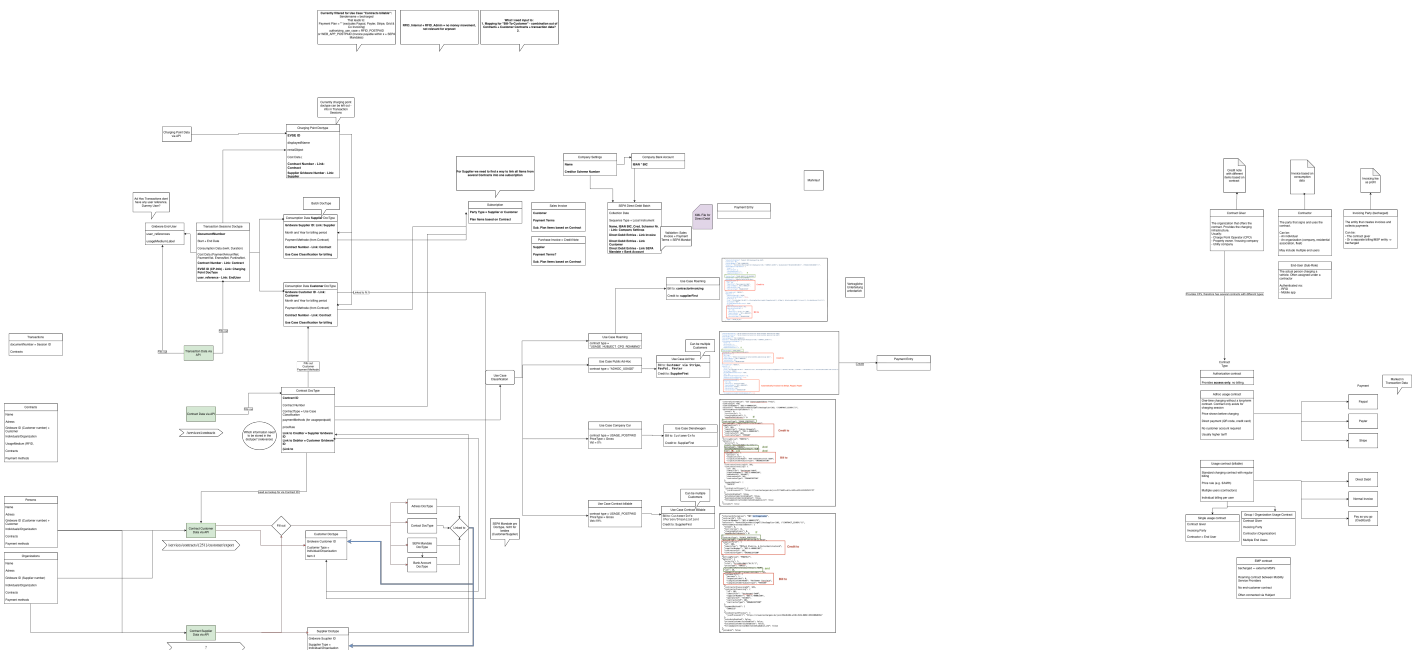
- Automatisches Importieren der CSV-Datei über Microsoft Power BI - **wurde ersetzt durch die Erlaubnis die API zu nutzen**
- Datenimport aus Gridware
 - Importieren, Anlegen und Verlinken von SEPA-Mandaten und Bankkonten
- Zahlungsabwicklung via SEPA
 - Automatisches Befüllen des SEPA Direct Debit DocTypes
 - Erstellung der SEPA XML-Dateien

IN AUSGIEBIGER PLANUNG Priorität 4 – Paket 4 ~20 std

Individualentwicklung

Ziel bis: Ende März

- Kompletter Stammdaten Datenimport aus Gridware **via API**



PLANUNG PRODUCTION RELEASE #2

Priorität 1 – Paket 3 ~20 std

Individualentwicklung

Bereitstellung bis: Ende Februar

- Automatisches Importieren der CSV-Datei über Microsoft Power BI - **wurde ersetzt durch die Erlaubnis die API zu nutzen**
- Datenimport aus Gridware
 - Importieren, Anlegen und Verlinken von SEPA-Mandaten und Bankkonten
- Zahlungsabwicklung via SEPA
 - Automatisches Befüllen des SEPA Direct Debit DocTypes
 - Erstellung der SEPA XML-Dateien

PLANUNG PRODUCTION RELEASE #3

Paket 4: Datenbasis ~70 Std.

Zeithorizont: bis April Ende/Mitte April

Übergeordnetes Ziel:

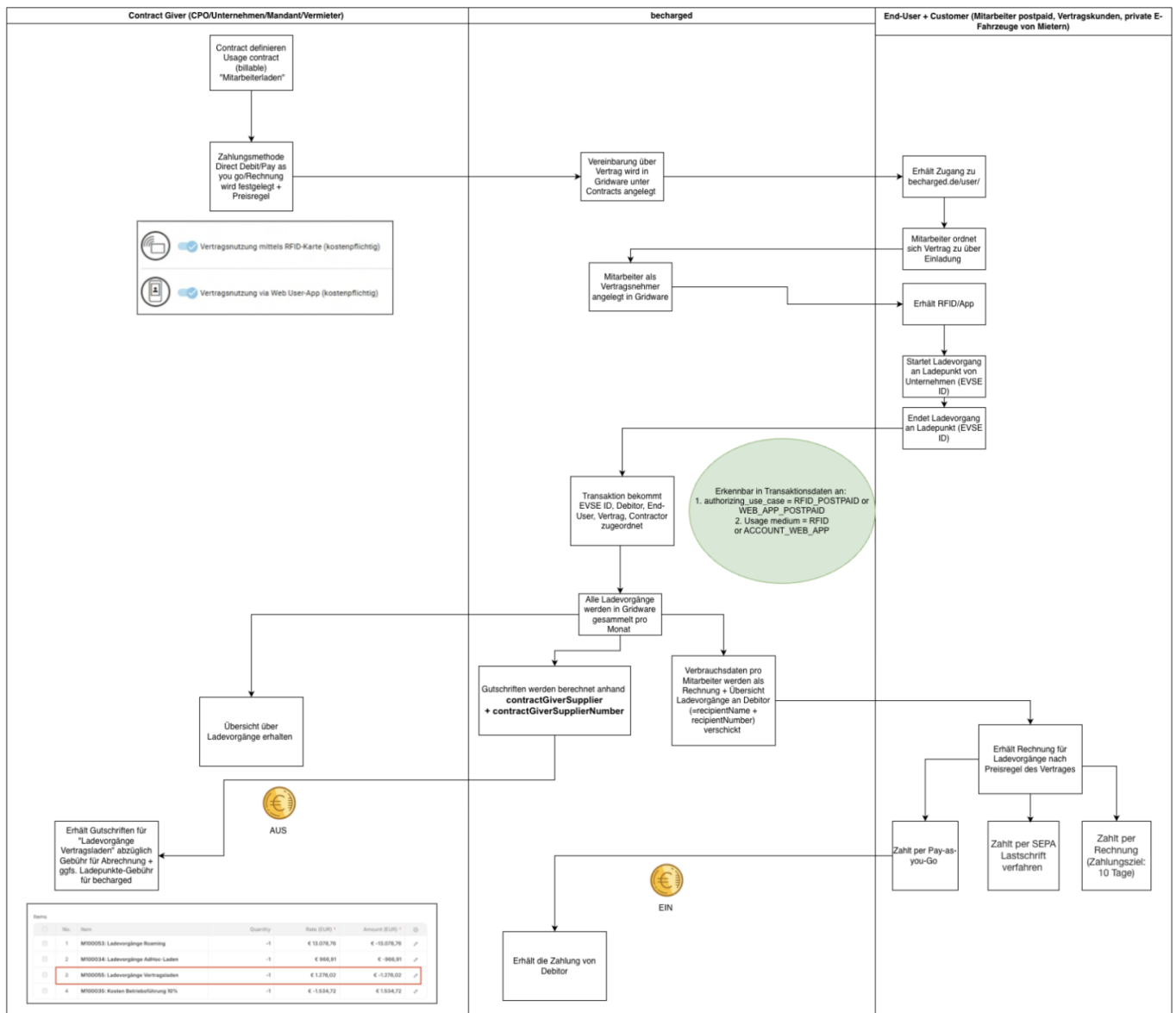
Datenbasis: Gridware → ERPNext über Swagger API stabil + sauberer Datenlayer (DocTypes/Mappings), sodass die späteren Use Cases darauf aufbauen können.

- **API-Import Basis steht**

- Swagger API angebunden (Auth, Rate-Limit/Retry, Logging)
- Staging-Import läuft reproduzierbar
- DoD: "Import Job" kann manuell/zeitgesteuert laufen lassen + schreibt saubere Logs
- **Stammdaten Import komplett**
 - Kunde, Lieferant, Kontakte, Adressen, SEPA Mandate, Bankaccounts, Vertrag
 - Lieferantenlogik über API abgebildet
 - DoD: Für N Testkunden: vollständige Stammdaten in ERPNext, inkl. Verlinkungen (Kontakt↔Kunde, Adresse↔Kunde, Bank↔Mandat etc.)
- **Transaktionsdaten Import**
 - Sessions/Transactions über Swagger API holen
 - Neuer DocType: "Single Session" (Transaktions-Session als Rohdatensatz)
 - Validierung + Mapping auf ERPNext Ziel-Doctypes
- **Neuer DocType „Vertrag“ als Data-Transfer + Billing-Anker**
 - Vertrag wie „wiederkehrender Auftrag“ gehandhabt
 - Lookup der Vertragskunden via Vertrag
 - DoD: Vertrag kann als Quelle dienen, um Stakeholder sauber abzuleiten (Kunde/Lieferant)
- **Use-Case Klassifizierung / Mapping-Regeln**
 - Use Cases klassifiziert
 - Mapping Kunde/Lieferant pro Use Case dokumentiert + implementiert
 - DoD: Für jede importierte Transaktion ist eindeutig: Use Case + beteiligte Parties + Vertrag/Zuordnung
- E-Invoice App?

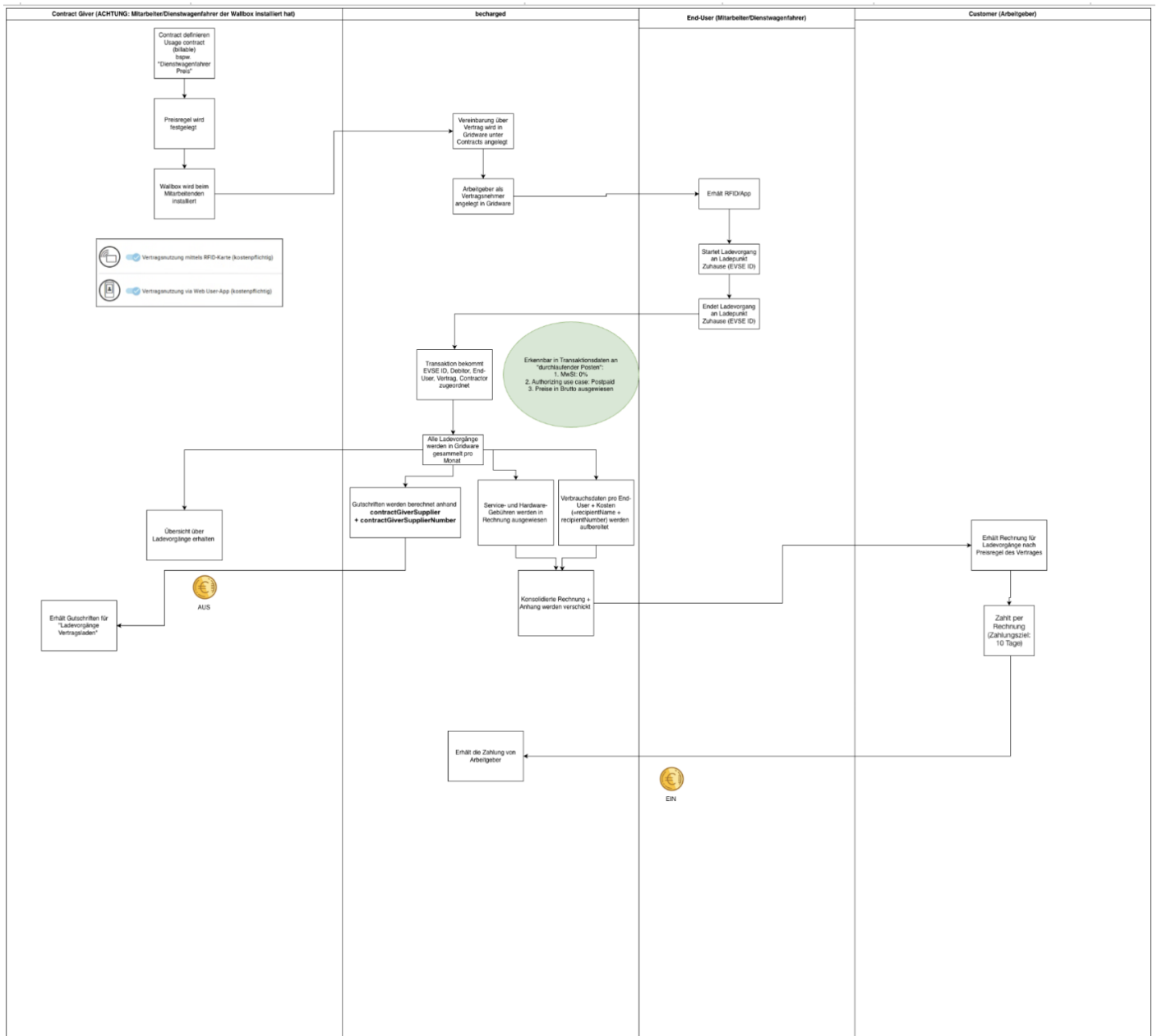
Paket 5: Use Case Grundlagen - ca. 50 Std.

- **Ziel:** „Datenmodell & Aggregationen“, auf denen alle Use Cases aufsetzen.
 - Single Sessions Erweiterung/ End-User DocTypes
 - DocType:Verbrauchsdaten Batch je Stakeholder (Kunde + Lieferant)
 - Use Case „Contract billable“ inkl. Gutschriften Lieferant mit neuer Logik abbilden
 - Vertrag: mehrere Artikel zuordenbar
 - Abonnement-Logik Kunde/Lieferant evaluieren/ersetzen
 - DoD: Für importierte Transaktionen des Use Cases "Contract billable" kann das System je Stakeholder Abrechnungs-/Auswertungsdaten generieren.



Paket 6: Use Case Dienstwagen - ca. 40 Std.

- **Ziel:** Durchlaufende Posten/Abrechnung und Reports Arbeitgeber/Arbeitnehmer
- Gutschriften und Sammel-Rechnungen
- **DoD:** Dienstwagen-Transaktionen werden korrekt zugeordnet und als Report/Abrechnung abrufbar für Arbeitgeber/Arbeitnehmer



Backlog:

Paket ?: Use Case Roaming -

- Ersetzen von Hubject?

Paket ?: Ladepunkte -

- Ladepunkte abbilden und mit Sessions und Contracts verbinden + Ladepunktgebühren

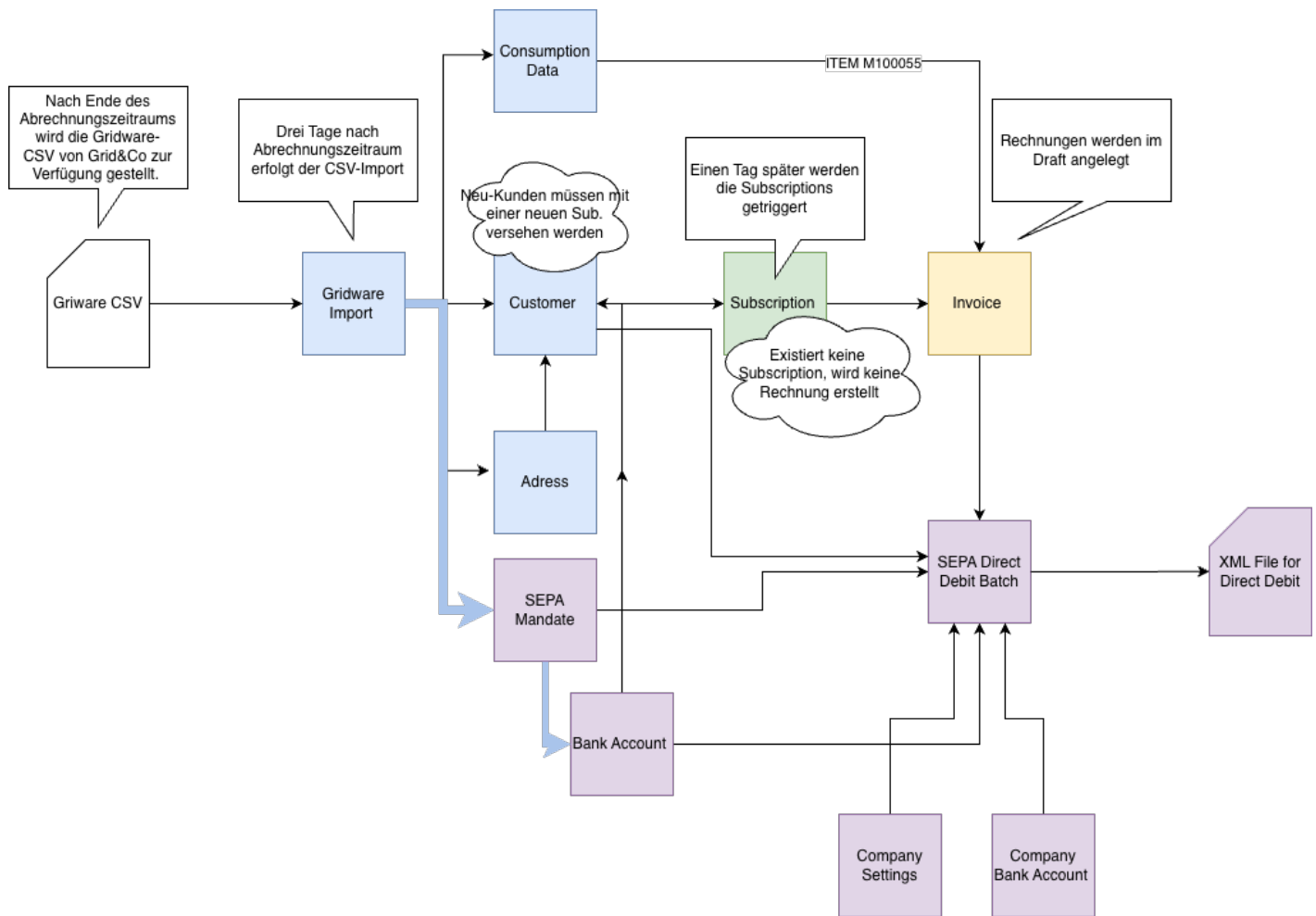
Paket?: Use Case AdHoc -

- Zahlungsverkehr in ERP Next abbilden, AdHoc Zahlungsmethoden mit Transaktionen + Belegen ablegen?
- Bank Transactions über PayPal - Kefiya

Paket ?: SIM Karten

Paket ?: Bestellung von Kundenladekarten

IST Stand 14.01.



	Aktueller Stand	Zu testen	Risiken, außerhalb unseres Bereiches
Gridware Import	- neue Kunden aktuell ohne Verbrauchsdaten - Adressen werden falsch verknüpft - PLZ und Ort im falschen Format - Import schlägt fehl mit originaler Gridware-CSV wegen Trennzeichen - SEPA-Mandate unvollständig / falsch zugeordnet	- Import von Verbrauchsdaten in DocType - Import von neuen Kunden - Import von Adressen - Verknüpfen Kunde mit Adresse - Import von SEPA-Mandaten - Erstellen von Bank Accounts - verknüpfen von SEPA-Mandaten, Bank Accounts, Kunde	- Es wird eine andere CSV Datei als die vereinbarte für den Upload verwendet (Entwicklung basiert auf API Schnittstellen CSV) - Die CSV enthält fehlerhafte Daten
Subscription	- aus Produktiv-System kopierte Sub. müssen deaktiviert werden da Error Log voll - ausführliches Testen notwendig mit Nov/Dez. Daten und Test Sub.	- automatisches Erstellen von Rechnungen mit richtigem Zeithorizont - korrekte Abrechnungszeiträume - Testfall 1: Nur Verbrauchsdaten - Testfall 2: Verbrauchsdaten mit anderem Artikel	- für (Neu)-Kunden werden keine Sub angelegt
Invoice	- Druckformat kleine Anpassungen erledigt - Anhang wird automatisch mit Rechnung verknüpft in ein Druckformat	- Rechnungen werden automatisch in Draft abgelegt - korrekte Verbrauchs- und Rechnungsdaten - Druckformate werden richtig angezeigt - Druckformat wird bei E-Mail Versand richtig verwendet	- MwSt. für Geschäftswagen wurden in Gridware noch nicht mit 0% ausgewiesen
SEPA Mandate/XML Datei	- DocType "Direct Debit" wurde angelegt - Direct Debit Entries müssen aktuell manuell angelegt werden, zukünftig über Filter - Notwendige Konfigurationen für Kreditoren-Infos notwendig	- automatisch werden richtige Kundentransaktionen für SEPA Lastschriftverfahren gefiltert - Kreditoren-Infos werden richtig aus Company extrahiert - Die XML Datei wird korrekt und lückenlos erstellt	- angewiesen auf das SEPA-Signatur Datum von Gridware

Abnahmetest auf Entwicklungssystem für Arbeitspaket #1

Ziel des Termins

- Abstimmung und Validierung des Imports von Verbrauchsdaten
- Prüfung der Subscription-Logik (manuell & automatisch)
- Sicherstellung korrekter Rechnungsstellung inkl. Anhänge und Versand

1. Import von Verbrauchsdaten

1. Prüfung: Verbrauchsdaten sind korrekt pro Kunde in Consumption Data DocType

2. Import von Neu-Kunden

1. Prüfung Neu-Kunden wurden alle angelegt und haben die Gridware ID zugeordnet

3. Subscriptions

1. Überblick Subscription-Logik

- Subscription wird **manuell** getriggert
- Subscription wird **automatisch** getriggert

1. Manuelles Triggern: Test-Case 1: Kunde hat Artikel "Ladevorgänge Vertragsladen" in Sub. angelegt
- Kunde CUST-2026-00020

<https://becharged.f.erpnext.com/app/subscription/ACC-SUB-2026-00001>

2. Manuelles Triggern: Test Case 2: Kunde hat zusätzlich mehr als einen Artikel in Sub. angelegt

- Kunde [CUST-2025-00063](#)

<https://becharged.f.erpnext.com/app/subscription/ACC-SUB-2025-00018>

3. Test Case 3 automatisches Triggern für morgen einrichten.

4. Liste mit bereits erstellten Rechnungen aus Subscription:

<https://becharged.f.erpnext.com/app/sales->

[invoice?creation=%5B%22Between%22%2C%5B%222025-12-](https://becharged.f.erpnext.com/app/sales-invoice?creation=%5B%22Between%22%2C%5B%222025-12-)

[31%22%2C%222025-12-](#)

[31%22%5D%5D&custom_consumption_data=%5B%22is%22%2C%22set%22%5D](#)

4. Rechnungen mit Anhang

1. Rechnungsdokument wird mit den richtigen Werte befüllt
2. Anhang fehlerfrei
 1. Besprechen Werte Verbrauch in kWh (siehe Screenshot) de-eng
3. Versand per Mail - wird das manuell von Sarah gemacht? - Anhang markieren

	Ladepunkt Name	Nutzung Start	Nutzung Ende	Nutzung Dauer	Nutzung (min)	Verbrauch (kWh)	Parkgebühren	Ladekosten	Fixkosten	Netto Preis	Nutzungsmedium
759	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-12 17:40:30	2026-01-12 17:52:00	11m 30s	11.50	7.124	0.0	1.13	0.0	1.13	RFID Fa. Wentowski 1
415	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-12 13:00:02	2026-01-12 13:54:13	54m 11s	54.18	24.144	0.0	3.86	0.0	3.86	RFID Fa. Wentowski 1
124	Chargebox: city-car-ka-li-2 Connector: 2	2026-01-12 12:13:40	2026-01-12 12:46:46	33m 6s	33.10	22.388	0.0	3.57	0.0	3.57	RFID CityCarKali
144	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-12 06:53:32	2026-01-12 06:56:26	2m 54s	2.90	0.0	0.0	0.0	0.0	0.0	RFID CityCarKali
388	Chargebox: city-car-ka-li-2 Connector: 2	2026-01-09 18:05:36	2026-01-09 18:09:28	3m 52s	3.87	0.0	0.0	0.0	0.0	0.0	RFID Fa. Wentowski 1
0	Chargebox: city-car-ka-li-2 Connector: 2	2026-01-09 17:11:40	2026-01-09 18:03:03	51m 23s	51.38	22.321	0.0	3.56	0.0	3.56	RFID Fa. Wentowski 1
0	Chargebox: city-car-ka-li-2 Connector: 2	2026-01-09 15:18:03	2026-01-09 15:26:40	8m 37s	8.62	3.764	0.0	0.61	0.0	0.61	RFID Fa. Wentowski 1
321	Chargebox: city-car-ka-li-2 Connector: 2	2026-01-09 12:31:15	2026-01-09 13:12:01	40m 46s	40.77	22.558	0.0	3.61	0.0	3.61	RFID Fa. Wentowski 1
764	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-09 09:49:03	2026-01-09 09:49:43	40s	0.67	105.0	0.0	0.02	0.0	0.02	RFID CityCarKali
558	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-09 09:18:47	2026-01-09 09:48:38	29m 51s	29.85	10.581	0.0	1.69	0.0	1.69	RFID CityCarKali
105	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-09 09:16:48	2026-01-09 09:18:03	1m 15s	1.25	0.0	0.0	0.0	0.0	0.0	RFID CityCarKali
581	Chargebox: city-car-ka-li-2 Connector: 1	2026-01-08 15:51:40	2026-01-08 15:52:08	28s	0.47	0.0	0.0	0.0	0.0	0.0	RFID Fa. Wentowski 1
0	Gesamt				238.55	217.88	18.05	0.00	0.00	18.05	

5. Plausibilität

- Aktive Subscriptions mit Artikel „Ladevorgänge Vertragsladen“
 - Erwartung: 42 Stück → Ist das korrekt?
 - <https://becharged.f.erpnext.com/app/subscription/view/list?plan=Ladevorg%C3%A4nge+Vertragsladen&status=%5B%22%21%3D%22%2C%22Cancelled%22%5D>
- Prüfung:
 - Kunden mit mehr als 1 Quantity von Item 100055 -> kommt das fachlich vor? Dieser Fall ist aktuell nicht abgesichert
 - <https://becharged.f.erpnext.com/app/subscription?qty=%5B%22%21%3D%22%2C%22Cancelled%22%5D&plan=Ladevorg%C3%A4nge+Vertragsladen>

becharged Aufgaben:

- Zugang zu Bank bereitstellen für Kefiya
- Neue Subscriptions anlegen
- Artikel alle auf

- Alle bestehenden Subscriptions auf das korrekte Start-/Abrechnungsdatum setzen
- Sehr wichtig: Gridware Customer Numbers manuell für bestehende Kunden ergänzen:
https://becharged.f.erpnext.com/app/customer?custom_gridware_customer_number=%5B%22is%22%2C%22not+set%22%5D
- Prüfung auf Duplikate in den Kundenstammdaten
- Einheitliche Benennung der Kunden:
 - Kundenname muss identisch mit dem Recipient Name in Gridware sein
 - Bei Import neuer Kunden zwingend prüfen, ob Kunde bereits existiert - wäre jedoch abgesichert, wenn Gridware ID schon angelegt ist

☐ Customer Name	Status	Customer Group	Territory	ID	2 of 2	♥
☐ Martin Buchall	Enabled	WEG Heusteig-L...	Alle Regionen	CUST-2024-00...	now	0 · ♥
☐ Martin Buchall	Disabled	WEG Heusteig-L...	Alle Regionen	CUST-2024-00...	2 M	2 · ♥

☐ Customer Name	Status	Customer Group	Territory	ID	2 of 2	♥
☐ Rademacher Steuer- und Unte	Enabled	Alle Kundengrup...	Alle Regionen	CUST-2025-00...	1 M	0 · ♥
☐ Gerd Rademacher	Enabled	Moser	Alle Regionen	CUST-2024-00...	1 y	0 · ♥

Project and Release Planung:

Release Rules

We effectively have two hard gates every month:

- **Gate A (1st of the month):** Consumption data must be imported cleanly and validated.
- **Gate B (7th of the month):** Billing/subscriptions/invoice run must complete reliably.

This means a “billing-safe” release must be in production before the 1st if it changes import/mapping — or at the latest before the 7th if it only affects billing logic.

Rule: “We deploy to production only if it does not risk the upcoming 1st/7th billing cycle.”

Default approach: Deploy **2-4 days before the 1st**, then only hotfixes.

Hotfix window: From the **1st to the 7th, fixes only, no new features**.

Phases:

Phase 1: Build (from the 8th to ~20th)

- Develop new features for the next billing cycle

Phase 2: Stabilization (approx. 21st to end of month)

- Bug fixing, performance improvements
- “No new features” except small, low-risk changes

Phase 3: Release Freeze + Cutover (last 3-5 days before the 1st)

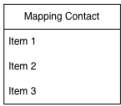
- Finalize migrations

Phase 4: Operational Monitoring (1st-7th)

- Monitor the import on the 1st (logs, counts)
- Monitor the invoice run on the 7th

Workshop - Next steps

- contract led
 - Contract defines relationships
 - sessions uses contract
- Goal till mid of April: Use Case "Contract billable" via API
- Steps:
 - clarify roles (customer, supplier, end-user)
 - Customer and supplier can be both?
 - New DocType to link customer and supplier
 - Define + create DocType
 - contract
 - single sessions
 - supplier
 - API endpoints + mapping
 - start import via API
 - contract or customer?
 - sessions
 - restructure consumption data to patch, collect via single sessions
- General integration:
 - architecture
 - logging
 - gridware integration
 - One place to manage all of the functionalities



- Scope:
 - log requests for API - trackability