immoGenio

No. 02
Juni 2026

BUCHUNGSREGEL

Buchungsregel-Engine in der Hausverwaltung: Wie wiederkehrende Belege automatisch SKR03-kontiert werden — und warum die Trefferquote die wichtigste Kennzahl ist

Buchungsregel-Engine: Pattern-Matching auf Lieferant, IBAN, Betrag und Verwendungszweck

AUTOR
ImmoGenio

VERÖFFENTLICHT
14. Juni 2026

ONLINE
www.immogenio.de/blog

— AUF EINEN BLICK

Inhalt

Wenn die Kontierung das eigentliche Engpass-Werkzeug wird

Eine Hausverwaltung mit 14 Mandanten verarbeitet pro Monat rund 320 Eingangsbelege: Strom-Allgemein-Rechnungen, Wasser- und Abwasser-Bescheide, Heizöl-Lieferungen, Hausmeister-Honorare, Versicherungsprämien, Wartungsverträge für Aufzüge, Brandschutz-Prüfprotokolle, kleinere Handwerker-Rechnungen. Der Buchhalter setzt sich an den Stapel, öffnet jeden Beleg, prüft Lieferant, Betrag, Mandantenzuordnung — und kontiert. Pro Beleg vergehen im Schnitt zwei Minuten. Hochgerechnet sind das elf Stunden pro Monat allein für die Kontierungs-Klicks. Davon entfallen rund 70 Prozent auf wiederkehrende Versorger- und Wartungsrechnungen, deren SKR03-Konto sich seit Jahren nicht geändert hat.

Diese 70 Prozent sind das Ziel einer Buchungsregel-Engine. Nicht die Sanierungs-Rechnung mit individueller Kostenstellen-Aufteilung, nicht die Sonderumlage mit Gesellschafter-Beschluss, nicht der überraschende Schadensfall — sondern die langweilige, vorhersehbare, wiederkehrende Buchung, deren Konto durch das Lieferanten-Profil bereits determiniert ist. Eine seriöse Engine löst diesen Engpass deterministisch, GoBD-konform und mit einer Trefferquote, die sich messen lässt.

Die Drei-Kategorien-Regel der Verwaltungs-Buchhaltung

Bevor eine Regel-Engine sinnvoll konfiguriert werden kann, muss klar sein, welche Belege sie überhaupt adressieren soll. In der Hausverwaltungs-Buchhaltung lassen sich Eingangsbelege grob in drei Kategorien einteilen.

Wiederkehrend, Konto stabil. Strom-Allgemein, Wasser, Heizöl, Hausmeister-Pauschale, Versicherungsbeiträge, Wartungsverträge. Lieferant ist konstant, das SKR03-Konto wechselt nicht. Diese Belege sind das Kerngebiet der Regel-Engine.

Wiederkehrend, Konto kontextabhängig. Hausgeld-Vorschüsse pro Eigentümer, Mietzahlungen pro Mietvertrag, Sonderzahlungen mit Verwendungszweck-Bezug. Hier ist nicht das SKR03-Konto die Frage — das ist meist 1200 (Forderungen) oder 1700 (Verbindlichkeiten) — sondern die Zuordnung zur Person und zum Vertrag. Diese Logik gehört in das Banking-Matching, nicht in die Buchungsregel-Engine.

Einmalig, individuell. Sanierungs-Rechnung mit Kostenstellen-Splitting, Sonderumlage mit Beschluss-Verweis, einmaliger Gutachter-Auftrag. Diese Belege erfordern manuelle Kontierung mit Kontext-Wissen — die Engine versucht hier gar nicht erst zu raten.

Diese Trennung ist nicht akademisch, sondern strukturierend: sie verhindert, dass die Engine in Bereichen automatisiert, in denen sie nicht zuverlässig sein kann.

Was eine Buchungsregel formal ist

Eine Buchungsregel ist eine logische Aussage der Form `WHEN <Bedingung> THEN <SKR03-Kontierung>`. Die Bedingung operiert auf einem definierten Set von Feldern, die entweder aus dem OCR-Output des Belegs oder aus dem SEPA-Verwendungszweck der Bank-Transaktion stammen.

- **Lieferant.** Name oder IBAN aus dem Beleg. Der Lieferant ist meist das stärkste Signal — eine Mainova-Rechnung ist mit hoher Wahrscheinlichkeit Strom-Allgemein.
- **Verwendungszweck.** Pattern aus dem Rechnungstext oder dem SEPA-Feld. Hier können Substring-Matches („Versicherungsprämie", „Wartung Aufzug", „Heizöl-Lieferung") oder regulärere Ausdrücke wirken.
- **Betrag.** Festbetrag (Pauschal-Honorar) oder Range (Strom zwischen 200 und 600 Euro pro Monat).
- **Beleg-Datum-Periode.** Saisonale Bedingungen — etwa „immer im Januar" für die jährliche Versicherungsprämie.
- **Quell-Konto.** Wenn der Mandant mehrere Bankkonten hat, kann die Engine das Bank-Konto in die Regel einbeziehen.

Die `THEN`-Seite definiert das SKR03-Konto, optional eine Kostenstelle, optional einen Steuersatz. Der Detail-Aufbau des Kontenplans — welche 4xxx-Konten in der Hausverwaltungs-Buchhaltung typisch sind — ist im SKR03-Artikel beschrieben.

Match-Strategie: Spezifitäts-Score statt First-Match

Die naive Implementation einer Regel-Engine arbeitet nach „erste Regel gewinnt". Das ist in der Buchhaltung nicht akzeptabel: zwei Regeln können legitim auf denselben Beleg zutreffen, und nur eine ist die fachlich richtige. Die Engine braucht daher einen Spezifitäts-Score.

Die Hierarchie ist intuitiv:

1. **IBAN + Betrag + Verwendungszweck** schlägt
2. **IBAN + Betrag** schlägt
3. **IBAN allein** schlägt
4. **Lieferantenname + Verwendungszweck-Exakt-Match** schlägt
5. **Lieferantenname allein** schlägt

6. **Verwendungszweck-Substring** schlägt
7. **Reines Betrags-Pattern**.

Eine Regel mit drei Bedingungen hat einen höheren Score als eine mit einer. Eine zeitlich begrenzte Regel („gültig bis 31.12.2026") schlägt die zeitlose Variante, weil sie spezifischer ist. Bei gleicher Anzahl Bedingungen entscheidet die Bedingungs-Stärke: IBAN ist stärker als Lieferantenname, Lieferantenname stärker als Verwendungszweck-Substring.

Konflikt-Behandlung: kein automatischer Buchungssatz im Zweifel

Wenn zwei Regeln dieselbe Spezifität haben — etwa zwei Lieferanten, die historisch dieselbe IBAN verwendet haben, oder zwei Verwendungszweck-Patterns mit identischer Länge — entsteht ein echter Konflikt. Eine seriöse Engine bucht in diesem Fall **nicht automatisch**.

Stattdessen legt sie den Beleg in einen Konflikt-Stapel mit beiden Vorschlägen und einem Hinweis: „Regel A schlägt Konto 4221 vor, Regel B schlägt Konto 4222 vor — bitte manuell entscheiden." Der Buchhalter wählt, und die Auswahl wird als zusätzliches Differenzierungs-Pattern für die Engine vermerkt. Beim nächsten Beleg, der sonst denselben Konflikt erzeugen würde, schlägt die Engine die zuvor gewählte Regel mit höherem Gewicht vor.

Das ist die zweite wichtige Eigenschaft: **die Engine eskaliert Unsicherheit, statt sie zu kaschieren.** Im Zweifel sieht der Buchhalter den Konflikt und entscheidet — was gleichzeitig dem GoBD-Vier-Augen-Prinzip gemäß BMF-Schreiben vom 28. November 2019 entspricht.

Lern-Effekt aus User-Korrektur: jeder Klick wird eine Regel

Die wichtigste UX-Mechanik einer praktikablen Regel-Engine ist die Ein-Klick-Regel-Erstellung aus Korrekturen. Wenn die Engine einen Beleg auf Konto A vorschlägt und der Buchhalter ihn auf Konto B korrigiert, erscheint nach der Buchung ein dezenter Hinweis: „Soll ich für diesen Lieferanten zukünftig Konto B verwenden?". Ein Klick — und eine neue Regel ist erstellt, mit dem Lieferanten als Bedingung und Konto B als Wirkung.

Das wirkt trivial, ist aber der Hebel, der eine Engine produktiv macht. Buchhalter konfigurieren keine Regeln in einem leeren Admin-Dialog. Sie korrigieren Vorschläge im Tagesgeschäft — und genau in diesem Moment muss die Engine fragen, ob die Korrektur eine einmalige Ausnahme oder ein Muster ist. Aus jedem Korrektur-Klick wird eine Regel, die Trefferquote steigt mit jedem Belegdurchgang.

Die Engine versteckt dabei keine Buchung und überschreibt keinen alten Buchungssatz. Bereits gebuchte Sätze bleiben unangetastet — Audit-Integrität schlägt Bequemlichkeit. Die neue Regel wirkt nur auf zukünftige Belege.

Trefferquote als wichtigste Kennzahl

Die einzige ehrliche Metrik einer Regel-Engine ist die Trefferquote: der Anteil eingehender Belege, der pro Mandant und pro Monat ohne manuelle Kontierung den Stapel passiert. Drei Stufen sind für die Telemetrie relevant.

- **Auto-Match.** Eine Regel matcht eindeutig, der Buchhalter bestätigt mit einem Klick.
- **Konflikt-Match.** Mehrere Regeln matchen, der Buchhalter entscheidet manuell zwischen den Vorschlägen.
- **No-Match.** Keine Regel greift, der Buchhalter kontiert frei und entscheidet, ob daraus eine neue Regel wird.

Bei 320 Belegen pro Monat ist eine Trefferquote von 65 bis 80 Prozent nach drei Monaten Eingewöhnung realistisch. Bei 80 Prozent Trefferquote sparen Sie pro Mandant rund 256 Belege mal 50 Sekunden Differenz pro Beleg — das sind etwa 13.600 Sekunden oder 3,8 Stunden pro Mandant pro Monat. Bei 14 Mandanten in der eingangs skizzierten Verwaltung sind das rund 53 Stunden — ein voller wöchentlicher Personenmonat, der nicht mehr für Klick-Arbeit verbrannt wird.

Wichtig ist, dass die Trefferquote pro Mandant gemessen wird, nicht aggregiert. Ein neuer Mandant startet bei 0 Prozent und braucht mehrere Buchungs-Zyklen, bis seine Lieferanten-Pattern in den Regelbestand eingeflossen sind. Wer nur den globalen Durchschnitt sieht, übersieht diese Anlauf-Phase.

Was die Engine bewusst nicht tut

Eine ehrliche Regel-Engine kennt ihre Grenzen.

- Sie **bucht nicht ohne Bestätigung**. Selbst bei 99 Prozent Konfidenz braucht der Buchungssatz einen menschlichen Klick — das ist die Vier-Augen-Forderung der GoBD und gleichzeitig die einzige Sicherung gegen automatisierte Fehlbuchungen, die im Extremfall den Tatbestand des § 263 StGB bei vorsätzlicher Manipulation streifen können.
- Sie **ändert keine bereits gebuchten Sätze rückwirkend**. Ein Buchungssatz ist nach AO § 145 und HGB § 257 ein Beweismittel — er darf korrigiert, aber nicht überschrieben werden.

- Sie **generiert keine neuen Lieferanten**. Wenn ein OCR-Beleg einen unbekannten Lieferanten extrahiert, eskaliert die Engine an den Buchhalter, der den Lieferanten manuell anlegt oder mit einem bestehenden zusammenführt. Automatisches Anlegen würde zu Dubletten führen, die später aufwendig konsolidiert werden müssen.
- Sie **operiert nur innerhalb eines Mandanten**. Cross-Tenant-Regel-Sharing — also „Ihre Regel funktioniert auch bei meiner Verwaltung" — ist in v1 bewusst nicht implementiert. Jede Verwaltung pflegt ihren eigenen Regel-Bestand, weil Lieferanten-Stamm und Konten-Konventionen sich zwischen Verwaltungen unterscheiden.

Verbindung zur OCR-Pipeline: Drei-Stufen-Workflow

Die Buchungsregel-Engine ist kein isoliertes System. Sie ist die zweite Stufe einer Pipeline, deren erste Stufe die OCR-Belegerfassung ist. OCR extrahiert aus dem PDF die Felder Lieferant, Belegdatum, Betrag, IBAN, Verwendungszweck. Die Regel-Engine konsumiert diese Felder und schlägt das Konto vor. Der Buchhalter bestätigt — und die dritte Stufe, der Buchungsstapel, übernimmt Soll/Haben.

Eine ehrliche UI zeigt alle drei Stufen mit ihrer jeweiligen Konfidenz. Beim OCR-Schritt steht etwa „Lieferant erkannt mit 94 Prozent Konfidenz", beim Regel-Schritt „Regel #14 matcht mit Spezifitäts-Score 87", beim Buchungs-Schritt „Soll 4221, Haben 1700, Steuersatz 19 Prozent". Wer eine dieser Stufen versteckt, behauptet eine Sicherheit, die das System nicht hat. Die Details der OCR-Schicht — welche Felder verlässlich extrahiert werden, welche Konfidenz-Schwellen sinnvoll sind — beschreibt der OCR-Artikel.

Praxis-Beispiel: WEG Beispielstraße 4

Eine Verwaltung betreut den Mandanten „WEG Beispielstraße 4". Drei Regeln sind im Bestand.

- **Regel #1.** Lieferant „Mainova AG" → SKR03 4221 (Strom Allgemein), Steuersatz 19 Prozent.
- **Regel #2.** IBAN „DE89 3704 0044 0532 0130 00" → SKR03 4222 (Wasser), Steuersatz 7 Prozent.
- **Regel #3.** Verwendungszweck enthält „Versicherungsprämie" und Lieferant „Allianz Versicherungs-AG" → SKR03 4360 (Versicherungen), Steuersatz 0 Prozent.

Beim 14. Beleg dieses Monats — eine Mainova-Stromrechnung über 340,18 Euro — durchläuft die Engine alle Regeln. Treffer #1 matcht mit Spezifitäts-Score 95, kein Konflikt. Im UI erscheint der Vorschlag: „Konto 4221 (Strom Allgemein), 340,18 Euro netto, 19 Prozent USt — Regel #1 vorgeschlagen". Der Buchhalter sieht das Belegbild, den Vorschlag, klickt

„Übernehmen". Im Buchungsstapel landet der Satz, beim DATEV-Export wird er Format-konform überführt. Die DATEV-Export-Mechanik ändert sich dadurch nicht — sie konsumiert den fertigen Buchungssatz.

Wie ImmoGenio das umsetzt

Im immogenio-Backend ist die Buchungsregel-Engine ein eigener fachlicher Block.

- **Tabelle** `buchungsregeln` in der Baseline-Migration. Spalten für Bedingungs-Felder (Lieferant, IBAN, Verwendungszweck, Betrag-Min, Betrag-Max, Datums-Range), für die Wirkung (Konto, Kostenstelle, Steuersatz), für Telemetrie (Trefferzähler, letzte Anwendung) und für Audit (erstellt von, geändert von).
- **Service** `buchungsregel-matcher.service.ts`. Implementiert den Spezifitäts-Score, die Konflikt-Erkennung und die Schnittstelle zur OCR-Output-Struktur. Reine Funktion ohne Seiteneffekte: gibt eine sortierte Liste von Regel-Treffern und einen Konflikt-Hinweis zurück. Die Persistenz des Buchungssatzes findet erst nach Bestätigung im UI statt.
- **Routen** `regeln.ts`. CRUD-Endpunkte für die Verwaltung des Regel-Bestands, plus ein `POST /regeln/:id/feedback` für die Korrektur-Lern-Mechanik. Lese- und Schreibzugriff sind über Permission-Gates abgesichert: nur Buchhalter und Tenant-Admin können Regeln anlegen oder ändern.
- **Verbindung zur OCR-Pipeline.** Die OCR-Migration 047 liefert das Schema, in dem extrahierte Beleg-Felder gespeichert werden. Der Matcher konsumiert dieses Schema direkt — keine Zwischen-Mapping-Schicht.
- **Verbindung zur Stapelverarbeitung.** Bestätigte Regel-Treffer werden in den Buchungsstapel überführt, der die Soll/Haben-Mechanik der doppelten Buchführung implementiert.

Verbindungen im immogenio-Stack

Die Regel-Engine ist Teil eines mehrstufigen Buchhaltungs-Stacks, dessen einzelne Schichten ineinandergreifen.

- **OCR-Belegerfassung.** Die Engine konsumiert OCR-Output. Welche Felder mit welcher Konfidenz extrahiert werden, beschreibt der OCR-Artikel.
- **SKR03-Kontenplan.** Regeln referenzieren SKR03-Konten. Welche 4xxx- und 8xxx-Konten in der Hausverwaltungs-Buchhaltung typisch sind, beschreibt der SKR03-Kontenplan-Artikel.
- **Doppelte Buchführung.** Die Engine schlägt nur Konten vor — Soll/Haben hält der Buchungsstapel. Die Mechanik beschreibt der Doppelte-Buchführung-Artikel.

- **Open Banking.** Banking-Transaktionen und Beleg-Buchungen werden bewusst getrennt verarbeitet. Wie das finAPI-Matching arbeitet, beschreibt der Open-Banking-Artikel.
- **Kreuzvalidierung gegen Vertragsdaten.** Bei Versicherungsbeiträgen und Wartungsverträgen lohnt der Abgleich mit der digitalen Vertragsmappe — Details im IDP-Kreuzvalidierungs-Artikel.

Grenzen der aktuellen Implementation

Mehrere Designentscheidungen sind bewusst konservativ.

- **Keine ML-basierte Klassifikation in v1.** Die Engine arbeitet deterministisch über Pattern-Matching, nicht über ein trainiertes Modell. Das bedeutet niedrigere theoretische Trefferquoten in unstrukturierten Fällen, aber vollständige Erklärbarkeit jeder einzelnen Buchungs-Empfehlung — was für GoBD-Audit und für das Buchhalter-Vertrauen wichtiger ist als ein Black-Box-Score.
- **Keine cross-tenant Regel-Vorlagen.** Jede Verwaltung pflegt ihren eigenen Regel-Bestand. Eine zentrale Vorlagen-Bibliothek mit „Top 50 Strom-Lieferanten in Deutschland" ist denkbar, aber in v1 nicht enthalten — sie erzeugt Konsolidierungs-Aufwand, der den Anfangs-Nutzen übersteigt.
- **Keine zeitlich auslaufenden Regeln mit Auto-Deaktivierung.** Wenn ein Wartungsvertrag 2027 endet, müssen die zugehörigen Regeln aktuell manuell deaktiviert werden. Eine zukünftige Version kann hier ein `gueltig_bis`-Feld auswerten.
- **Keine Mehrfach-Kontierung in einer Regel.** Eine Regel matcht auf ein Konto. Belege mit Splitting (Sanierung mit anteiliger Kosten-Trennung) werden bewusst nicht regelbasiert behandelt — sie gehören in die manuelle Kategorie.

Diese Grenzen sind dokumentiert, nicht versteckt. Eine Engine, die behauptet, alle Belege automatisch zu kontieren, ist entweder falsch implementiert oder unehrlich beworben.

Wo wir stehen

Die Buchungsregel-Engine ist im immogenio-Backend produktiv. Tabelle, Service, Routen, Konflikt-Auflösung und die Ein-Klick-Regel-Erstellung aus User-Korrekturen laufen. Trefferquoten werden pro Mandant geloggt und im Buchhaltungs-Cockpit als Zeitreihe sichtbar gemacht. Die Verbindung zur OCR-Pipeline und zum Buchungsstapel ist hergestellt. Was noch aussteht, ist die zeitliche Begrenzung von Regeln und die Auswertung von saisonalen Mustern über mehrere Buchungsjahre.

Kontakt

Wenn Sie als Hausverwalter, Buchhalter oder Steuerberater Ihre wiederkehrenden Buchungen automatisieren wollen, ohne die Kontrolle aus der Hand zu geben, sprechen Sie uns an. Wir zeigen Ihnen, wie sich aus Ihren Korrektur-Klicks im Tagesgeschäft schrittweise ein Regelbestand aufbaut, der nach drei Monaten 70 bis 80 Prozent Ihrer Eingangsbelege automatisch korrekt SKR03-kontiert — vollständig im Sinne der GoBD, mit nachvollziehbaren Vorschlägen und einer Trefferquote, die Sie im Cockpit jederzeit messen können.

Erreichbar unter kontakt@immogenio.de.