

IDP



IDP für Mietverträge und Versicherungspolicen: Wie eine ehrliche KI-Pipeline aus PDFs strukturierte Daten macht — mit Konfidenz pro Feld

Intelligent Document Processing in der Hausverwaltung:
Klassifizierer, typ-spezifische Extraktion, Konfidenz pro
Feld

AUTOR

ImmoGenio

VERÖFFENTLICHT

26. Juni 2026

ONLINE

www.immogenio.de/blog

Inhalt

- 01 Der Bestandsübergang, der drei Wochen kostet
 - 02 Was IDP von OCR unterscheidet
 - 03 Drei Phasen einer ehrlichen IDP-Pipeline
 - 04 Konfidenz pro Feld als wichtigste Disziplin
 - 05 Auto-Approve-Thresholds
 - 06 Cross-Validation als zweite Sicherheitsstufe
 - 07 Few-Shot-Lernen pro Tenant — ohne Fine-Tuning
 - 08 Praxisbeispiel: ein Mietvertrag im Live-Lauf
 - 09 Wie ImmoGenio das technisch umsetzt
 - 10 DSGVO: warum HITL keine Compliance-Schikane ist
 - 11 Verbindungen zu den Nachbar-Modulen
 - 12 Grenzen — was v1 ehrlich nicht kann
-

13 Wo ImmoGenio steht

14 Wenn Sie Bestände übernehmen

Der Bestandsübergang, der drei Wochen kostet

Eine Mietverwaltung übernimmt einen neuen Bestand mit 87 Wohneinheiten. Die Übergabe besteht — wie üblich — aus einem Ordner voller PDFs: 87 Mietverträge, 14 Versicherungspolicen, 23 Wartungsverträge, 9 Heizkostenabrechnungen aus dem Vorjahr. Der Bestandsübergangs-Workflow verlangt für die Ist-Aufnahme pro Mietvertrag rund acht bis zwölf Felder — Mieter-Stammdaten, Vertragsbeginn, Kaltmiete, Nebenkostenvorauszahlung, Kautions, Heiz-Schlüssel, Mietsicherheit, Kündigungsfristen. Manuell abgetippt sind das drei Wochen Arbeit, in denen die Buchhaltung niemand macht.

Das Problem ist nicht, dass es niemand kann. Das Problem ist, dass kein Mensch acht Stunden am Stück IBANs aus Versicherungspolicen abschreiben sollte. Genau hier setzt Intelligent Document Processing an — wenn man es ehrlich baut.

Was IDP von OCR unterscheidet

OCR — Optical Character Recognition — beantwortet die Frage „Welche Zeichen stehen in diesem Bild“. Mehr nicht. Sie bekommen einen Textstrom zurück und dürfen sich selbst überlegen, welche Zeile davon der Vermietername ist und welche die Adresse.

Intelligent Document Processing geht drei Schritte weiter. Erstens **klassifiziert** es das Dokument typ-spezifisch — Mietvertrag, Versicherungspolice, Wartungsvertrag, Rechnung, Heizkostenabrechnung. Zweitens **extrahiert** es Felder gegen ein typ-spezifisches JSON-Schema, mit einer Konfidenz pro Feld. Drittens **kreuz-validiert** es das Ergebnis gegen Ihre Stammdaten und schickt unsichere Felder in einen Human-in-the-Loop-Review, dessen Korrekturen den nächsten Lauf besser machen.

Das ist der Unterschied zwischen „der Computer kann lesen“ und „der Computer hat verstanden, dass das ein Mietvertrag ist und welche Felder darin stecken müssen“.

Drei Phasen einer ehrlichen IDP-Pipeline

Phase 1: Klassifizierer

Bevor irgendetwas extrahiert wird, muss klar sein, was vorliegt. Der Klassifizierer beantwortet eine einzige Frage: Welcher Dokumenttyp ist das? Mietvertrag, Versicherungspolice, Wartungsvertrag, Eingangsrechnung, Heizkostenabrechnung des Dienstleisters, Mietkautionsbestätigung der Bank. Die Antwort kommt als Label plus Konfidenz.

Der Pre-Filter davor erkennt strukturierte Eingaben — eine ZUGFeRD-PDF mit eingebetteter XML oder eine reine XRechnung — und übergibt sie direkt an den E-Rechnungs-Parser. Der Klassifizierer arbeitet ausschließlich am Unstrukturierten, weil dort die KI gebraucht wird. Wer das durcheinanderwirft, lässt eine teure Bild-KI strukturierte XML-Daten erraten, die deterministisch parsbar wären. Mehr dazu im Beitrag zur E-Rechnungs-Pflicht und dem In-Flight-Swap zwischen XRechnung und ZUGFeRD.

Phase 2: Typ-spezifische Extraktion

Pro Dokumenttyp existiert ein definiertes JSON-Schema mit Pflichtfeldern. Für einen Mietvertrag nach § 535 BGB sieht das vereinfacht so aus:

```
{
  "vermieter": { "name": "string", "anschrift": "string" },
  "mieter": [{ "name": "string", "geburtsdatum": "date" }],
  "wohnung": { "anschrift": "string", "wohnflaeche_qm": "number", "lage": "string" },
  "mietbeginn": "date",
  "mietende_befristet": "date | null",
  "kaltmiete": "number",
  "nebenkostenvorauszahlung": "number",
  "heizkostenvorauszahlung": "number",
  "kaution_betrag": "number",
  "kaution_form": "barzahlung | buergschaft | sparbuch",
  "kuendigungsfrist_monate": "number"
}
```

Für eine Versicherungspolice ein anderes Schema — Versicherer, Vertragsnummer, versicherte Sache, Sparte, Jahresprämie, Selbstbehalt, Kündigungsfrist, Hauptfälligkeit. Für einen Wartungsvertrag wieder ein anderes — Auftragnehmer, Anlagenbezeichnung, Wartungsintervall, Pauschale, Kündigungsfrist, AVB-Stand.

Die Trennung ist wichtig, weil ein Versicherer, der bei einem Mietvertrag im Schema „Vermieter“ zugewiesen wird, semantischer Unsinn ist. Typ-spezifische Schemata zwingen das Modell, die richtigen Fragen zu stellen.

Phase 3: Kreuzvalidierung gegen Stammdaten

Die KI liefert einen Vorschlag. Der Vorschlag wird gegen Ihren Datenbestand geprüft. Eine Lieferanten-IBAN aus einer Versicherungspolice wird mit der `lieferanten_bankverbindungen`-Whitelist abgeglichen — gibt es die IBAN bereits, ist gut. Ist sie neu, wird das Feld als „validierungspflichtig“ markiert, auch wenn die KI-Konfidenz hoch war.

Ein Mieter-Name wird gegen den Mieter-Stammdatensatz abgeglichen, sobald einer existiert. Eine Kaltmiete wird gegen eine marktübliche Bandbreite des Objekts geprüft — bei extremen Abweichungen, etwa 12.000 Euro statt 1.200 Euro, setzt die Pipeline ein Anoma-

ly-Flag. Nicht, weil 12.000 Euro unmöglich wären, sondern weil das eine Frage wert ist.

Konfidenz pro Feld als wichtigste Disziplin

Die häufigste Schwäche schlechter IDP-Implementierungen ist eine Konfidenz auf Beleg-Ebene. Ein einziger Score von 0,87 für „dieser Vertrag wurde extrahiert“ sagt Ihnen nichts. Vielleicht stimmen Vermieter, Mieter und Anschrift mit jeweils 0,99 – und die Kaltmiete wurde mit 0,42 geraten.

Brauchbare Steuerung liefert nur die Konfidenz pro Feld. Ein Score zwischen 0,0 und 1,0 für jedes einzelne Datenfeld. Lieferant 0,98, Kaltmiete 0,93, IBAN 0,82 – die IBAN landet im Review, alles andere darf durchlaufen. Das ist der Unterschied zwischen „87 Verträge müssen alle nachgeprüft werden“ und „in 87 Verträgen sind 14 Felder unsicher, prüfen Sie diese 14“.

Auto-Approve-Thresholds

Pro Dokumenttyp und Feld konfiguriert die Pipeline Schwellwerte, ab denen ein Feld als bestätigt durchläuft. Konservative Defaults für die Mietverwaltung sehen ungefähr so aus:

FELD	THRESHOLD	BEGRÜNDUNG
lieferant_iban	0,95	Zahlungsrelevant – eine falsche IBAN ist Geldverlust
kaltmiete	0,90	Buchungsrelevant, aber durch Hausgeld-Cross-Check absicherbar
kautionsbetrag	0,95	Treuhand-relevant, getrennte Anlage erforderlich
mietbeginn	0,90	Stichtag für viele Folgen, aber meist klar formatiert
wohnflaeche_qm	0,85	Schätzgrundlage, niedrige Toleranz unkritisch

Felder unter dem Threshold gehen in den HITL-Review-Stapel. Felder darüber dürfen ohne Sichtprüfung in den Stammdatensatz wandern – nicht, weil die KI unfehlbar wäre, sondern weil ein Mensch, der 87 Mal „passt“ klickt, am Ende des Tages auch nichts gesehen hat. Stichprobe und Audit-Trail ersetzen die rituelle Bestätigung.

Cross-Validation als zweite Sicherheitsstufe

Auch ein KI-Score von 0,98 reicht nicht, wenn die Substanz fehlt. Eine Lieferanten-IBAN, die das Modell mit 0,98 ausgibt, wird trotzdem manuell bestätigt, falls sie noch nicht in der Whitelist steht. Tauchen für denselben Lieferanten plötzlich zwei IBANs auf, ohne dass die alte aktiv widerrufen wurde — ein typisches Muster bei Rechnungs-Manipulation —, gilt zwingend HITL-Pflicht. Diese Logik teilt sich die IDP-Pipeline mit der Anti-Swap-Mechanik der E-Rechnungs-Verarbeitung.

Die Logik ist in beiden Fällen: KI-Konfidenz und Stammdaten-Konfidenz sind zwei unabhängige Achsen. Die KI sagt, wie sicher sie ist, was im Dokument steht. Die Stammdaten sagen, ob das, was im Dokument steht, zur bisherigen Geschäftsbeziehung passt. Erst beide zusammen ergeben Auto-Approve.

Few-Shot-Lernen pro Tenant — ohne Fine-Tuning

Wenn der Buchhalter im HITL-Review ein Feld korrigiert — die Pipeline hatte bei der Hauptfälligkeit „01.07.“ extrahiert, korrekt war „01.04.“ —, wird die Korrektur als Trainingsbeispiel gespeichert. Beim nächsten Extraktionslauf desselben Tenants wandert dieses Beispiel als Few-Shot-Demonstration in den Prompt: „Bei diesem Mandanten wurde ein ähnliches Dokument so extrahiert.“ Das Modell sieht das Muster und passt seine Antwort an.

Wichtig: Das ist **kein Fine-Tuning des Modells**. Es gibt keine Modell-Versions-Verwaltung, keine GPU-Trainingsläufe, keine zentral gespeicherten Mandanten-Daten in Modellgewichten. Es ist Prompt-Injection mit tenant-spezifischen Beispielen aus der `idp_few_shots`-Tabelle, die der `idp-fewshot-builder.service.ts` zur Laufzeit zusammenstellt.

Drei Vorteile dieser Architektur:

- **Sauberer Tenant-Cut:** Korrekturen einer Verwaltung verbessern nur deren eigene Pipeline. Cross-Tenant-Lernen ist baulich ausgeschlossen — relevant für Privacy by Design nach Art. 25 DSGVO.
 - **Sofort wirksam:** Eine Korrektur am Montagvormittag verändert die Extraktion am Montagnachmittag. Kein Trainingslauf, keine Deployment-Pipeline.
 - **Reversibel:** Falsch markierte Few-Shots lassen sich einzeln deaktivieren. Bei einem Fine-Tune wäre das ein Re-Train.
-

Praxisbeispiel: ein Mietvertrag im Live-Lauf

Ein Mietvertrag als PDF, fünf Seiten, vom Vermieter eigenhändig in Word formatiert. Der Pre-Filter prüft auf XML-Anhänge — keine. Übergabe an den Klassifizierer.

Klassifizierer: "Mietvertrag"

Konfidenz 0,99

Typ-spezifische Extraktion mit dem Mietvertrags-Schema:

vermieter	0,98	Müller Immobilien GmbH
mieter[0]	0,97	Anna Schmidt
wohnung.anschrift	0,94	Musterstraße 12, 10115 Berlin
mietbeginn	0,99	2026-08-01
kaltmiete	0,93	1.200,00 EUR
nebenkostenvorauszahlung	0,92	240,00 EUR
heizkostenvorauszahlung	0,91	120,00 EUR
kautions_betrag	0,87	3.600,00 EUR
kautions_form	0,95	Barzahlung

Kautions-Score 0,87 unter dem Threshold von 0,95 — Feld geht in den HITL-Review. Der Buchhalter öffnet das `IdpReviewSheet`, sieht links das PDF auf Seite 3, rechts das extrahierte Feld „3.600,00 EUR“. Im PDF steht „Mietsicherheit: 3.600 Euro (drei Monatskaltmieten)“. Bestätigung in einem Klick. Die Korrektur — in diesem Fall „Wert war richtig, Kontext ‚drei Monatskaltmieten‘ als Beleg“ — wandert in `idp_few_shots` und stützt zukünftige Extraktionen mit ähnlicher Formulierung.

Aus zwölf Minuten manueller Eingabe pro Vertrag werden 90 Sekunden HITL-Review für die unsicheren Felder. Bei 87 Verträgen mit durchschnittlich 1,5 unsicheren Feldern reduziert sich der Aufwand von rund 17 Stunden auf etwa zwei Stunden — bei höherer Datenqualität, weil die Stammdaten-Kreuzvalidierung Tippfehler abfängt, die ein müder Mensch um 16 Uhr nicht mehr sieht.

Wie ImmoGenio das technisch umsetzt

Migration 071 legt das Datenmodell an: `idp_extraktionen` als Hauptobjekt mit Eingabe-Datei, Klassifikator-Ergebnis, Schema-Version und JSON-Result; `idp_feldkorrekturen` als Append-Only-Log jeder HITL-Korrektur mit Vorher-Nachher-Wert und Begründung; `idp_few_shots` als Tenant-isolierte Beispielsammlung für den Prompt-Builder. Migration 072 ergänzt einen Soft-Link `belege.idp_extraktion_id`, sodass jeder Beleg seine Extraktions-Herkunft kennt.

Die Service-Schicht teilt Verantwortung sauber auf:

- `idp-prefilter.service.ts` — erkennt strukturierte Eingaben und routet an den E-Rechnungs-Parser
- `idp-classifier.service.ts` — Dokumenttyp-Erkennung
- `idp-extractor.service.ts` — typ-spezifische Schema-Extraktion gegen `pixtral-large-latest`
- `idp-confidence.service.ts` — Threshold-Auswertung pro Feld
- `idp-crossvalidation.service.ts` — Stammdaten-Abgleich
- `idp-feedback.service.ts` — HITL-Korrekturen entgegennehmen
- `idp-fewshot-builder.service.ts` — tenant-spezifische Beispiele in den nächsten Prompt einfügen

Die Routen liegen unter `/api/idp/extraktionen`. Die UI-Schicht liefert `IdpView` als Cockpit-Übersicht mit Filter auf „im Review“, `IdpReviewSheet` mit feldweisen Konfidenz-Ampeln und PDF-Side-by-Side, `IdpStartButton` als Row-Action in den Dokumenten-Listen und `MietvertragIdpAssistant` direkt im Mietvertrag-Modal — falls der Verwalter einen Vertrag hochlädt, schlägt die KI Felder vor, die der Verwalter mit einem Klick übernimmt.

DSGVO: warum HITL keine Compliance-Schikane ist

Vier rechtliche Pfeiler, die in der IDP-Architektur baulich verankert sind:

Art. 22 DSGVO — kein ausschließlich automatisiertes Entscheiden. Eine Auto-Approve-Pipeline, die einen Mietvertrag ohne menschliche Beteiligung in den Stammdaten anlegt und automatisch eine Mahnung erzeugt, wäre ein Risikofall. ImmoGenio macht aus IDP grundsätzlich einen **Vorschlag**, der spätestens an einer rechtlich relevanten Stelle — Vertragsabschluss, Zahlung, Kündigung — eine menschliche Bestätigung erfordert. Stammdaten-Anlage zählt dazu.

Art. 5 lit. c — Datenminimierung. Die typ-spezifischen Schemata extrahieren nur Felder, die für den Zweck erforderlich sind. Ein Mietvertrag enthält oft den Geburtsort des Mieters — wenn die Hausverwaltung diesen für ihre Aufgaben nicht braucht, gehört er nicht ins Schema, also auch nicht in den Stammdatensatz.

Art. 25 — Privacy by Design. Tenant-Cut der Few-Shots ist kein Feature, das später nachgerüstet wird, sondern Schema-Voraussetzung der `idp_few_shots`-Tabelle. Cross-Tenant-Lernen ist baulich ausgeschlossen.

Art. 28 — Auftragsverarbeitung. Mit Mistral als KI-Provider besteht ein AVV. Das Hosting läuft in der EU (Frankreich). Vertraglich ausgeschlossen ist die Nutzung der Mandanten-Daten für Modell-Training auf Provider-Seite — die Few-Shots bleiben in der ImmoGenio-Datenbank, der Provider sieht sie nur transient während des einzelnen Calls.

Hinzu kommt die GoBD-Linie nach BMF-Schreiben vom 28.11.2019: Belegfunktionsfähigkeit, Unveränderbarkeit, Nachvollziehbarkeit. IDP-Extraktionen sind selbst keine Belege, sondern abgeleitete Daten — der Original-Beleg bleibt revisionssicher gespeichert, jede Korrektur wird im Audit-Trail mit Zeitstempel und Bearbeiter erfasst, der Aufbewahrungspflicht nach § 147 AO ist Genüge getan.

Verbindungen zu den Nachbar-Modulen

IDP ist kein Solitär. Es lebt im Zusammenspiel mit:

- der **OCR-Belegerfassung**: OCR ist der Spezialfall „Eingangsrechnung“. Strukturell ist es derselbe Mechanismus — Klassifizierer, Schema, Konfidenz, HITL —, nur mit einem fixen Dokumenttyp und einem Mobile-Capture-Pfad. Details im Beitrag zur OCR-Belegerfassung mit Mistral Vision.
- der **Buchungsregel-Engine**: Sobald IDP eine Eingangsrechnung extrahiert hat, läuft das Ergebnis durch die Regel-Engine, die Lieferant, Verwendungszweck und Betrag auf SKR03/SKR04-Konten kontiert. IDP liefert den Input, die Regel-Engine den Buchungssatz — siehe Buchungsregel-Engine für die automatische Kontierung.
- dem **DATEV-Export**: Belege, die aus IDP entstanden sind, tragen ihre Extraktions-Herkunft im Belegbundle für den Steuerberater — siehe DATEV-Export Format 7 mit Belegbundle.
- dem **Mieter-Workflow**: Mietvertrags-Daten landen über den `MietvertragIdpAssistant` direkt im `mietvertraege`-Stammdatensatz — der Verwalter prüft, der Datensatz ist da.

Grenzen — was v1 ehrlich nicht kann

Drei Einschränkungen, die in der Mandanten-Kommunikation klar genannt werden sollten:

- **Keine handschriftlichen Verträge**: Pixtral leistet bei Druckschrift sehr gute Arbeit, bei Handschrift sinkt die Konfidenz dramatisch. Wer einen handschriftlichen Mietvertrag aus den Achtzigern hat, tippt ihn ab.
- **Keine komplexen Tabellen-Hierarchien**: Eine flache Tabelle „Wohnung — Lage — Fläche — Miete“ funktioniert. Eine verschachtelte Heizkostenabrechnung mit Spalten-Spans und Zwischensummen ist eine separate Disziplin, die mit Tabellen-Detection-Modellen gelöst wird, nicht mit dem allgemeinen Klassifizierer.
- **Nur Deutsch**: Die Schemata, die Few-Shots und die Validierungs-Heuristiken sind auf deutschsprachige Verträge ausgerichtet. Englische oder französische Verträge würden funktionieren, aber ohne die tenant-spezifischen Beispiele wäre die Konfidenz niedriger.

Diese Grenzen sind nicht Architektur-Schwächen, sondern bewusste Scope-Schnitte für v1. Sie reduzieren die Komplexität der Pipeline und halten den HITL-Aufwand im Bereich des wirtschaftlich Sinnvollen.

Wo ImmoGenio steht

Das IDP-Epic 102 ist abgeschlossen und produktiv. Die Migrationen 071 und 072 sind ausgerollt, die Service-Schicht ist im API-Container live, das Modell `pixtral-large-latest` ist über den AVV-gedeckten Mistral-Endpunkt angebunden, die UI-Komponenten sind in den Dokumenten-Listen und im Mietvertrag-Modal verankert. Tenant-spezifische Few-Shots werden seit dem Rollout gesammelt und gehen mit jedem Lauf in den Prompt.

Die Beobachtung aus den ersten Mandanten: Die Konfidenz pro Feld steigt nach den ersten 20 bis 30 HITL-Korrekturen pro Dokumenttyp messbar — bei Mietverträgen vor allem bei den Feldern, die der jeweilige Vermieter eigenwillig formatiert (Kautions als „Mietsicherheit“, „Betriebskosten“ statt „Nebenkosten“). Die Pipeline lernt die Sprache des Bestands, nicht die generische Vertragssprache.

Wenn Sie Bestände übernehmen

Wenn Sie regelmäßig Bestandsübergänge machen, lohnt sich ein Gespräch. Wir zeigen Ihnen den Workflow an einem Ihrer Verträge — Sie sehen, welche Felder mit welcher Konfidenz extrahiert werden, wo die Schwellen sinnvoll liegen und wie der HITL-Review aussieht. Schreiben Sie uns an kontakt@immogenio.de oder buchen Sie eine Demo. Drei Wochen Eintipp-Arbeit sind kein Naturgesetz.