

GENIOFLOW

GenioFlow — Die Evolution der Immobilienverwaltung: Geführte Workflows statt freier Statusfelder

GenioFlow steuert Mahnwesen, Mieterhöhung nach § 558 BGB und Instandsetzungs-Freigaben als explizite Zustandsmaschinen.

AUTOR

ImmoGenio

VERÖFFENTLICHT

12. April 2026

ONLINE

www.immogenio.de/blog

Inhalt

- 01 Das Problem mit freien Statusfeldern

- 02 State Machines: der theoretische Unterbau

- 03 Die drei aktuellen Workflow-Typen

- 04 Nebenläufigkeit und Skalierung: Was passiert bei gleichzeitigen Zugriffen?

- 05 Compliance-Aspekte: Was der Audit-Trail für den Verwaltungsalltag bedeutet

- 06 Rollout ohne Risiko: Feature-Flag und Backfill

- 07 Warum das der entscheidende Reifesprung ist

W

ER SEIT MEHR ALS FÜNF JAHREN EINE HAUSVERWALTUNG BETREIBT, KENNT DIE STILLE GEFAHR des freien Statusfelds. Ein Datenbankfeld mit dem Wertebereich `entwurf`, `versandt`, `bezahlt`, `storniert` klingt sauber – und ist es auf den ersten Blick auch. Das Problem entfaltet sich erst im Betrieb: Die Software lässt Sie eine Mahnung als `versandt` markieren, bevor das PDF überhaupt generiert wurde. Sie lässt Sie den Status auf `bezahlt` setzen, auch wenn kein Zahlungseingang verbucht ist. Sie lässt Sie eine Mieterhöhung nach § 558b BGB versenden, bevor die gesetzliche Dreimonats-Frist abgelaufen ist. Die Verantwortung liegt beim Verwalter, und damit auch das Haftungsrisiko.

GenioFlow, die Workflow-Engine von ImmoGenio, geht einen anderen Weg. Anstatt Status als freies Feld zu speichern, modelliert sie jeden Verwaltungsprozess als endlichen Automaten: Zustände, erlaubte Übergänge und Bedingungen. Was nicht explizit erlaubt ist, ist verboten – und wird serverseitig abgewiesen, unabhängig davon, was die Benutzeroberfläche anzeigt.

Das Problem mit freien Statusfeldern

Freie Statusfelder haben einen offensichtlichen Vorteil: Sie sind einfach zu implementieren. Eine Spalte in der Datenbank, ein Select-Dropdown im Frontend – fertig. Der Nachteil ist subtiler, aber schwerwiegend.

Ein Statusfeld speichert einen Zustand, aber nicht die erlaubten Übergänge zwischen Zuständen. Es kann nicht wissen, ob der Übergang von `entwurf` nach `bezahlt` überhaupt zulässig ist (er ist es nicht – eine Mahnung muss erst versendet worden sein, bevor sie als bezahlt markiert werden kann). Es kann nicht überprüfen, ob eine Bedingung erfüllt ist (eine dritte Mahnstufe darf erst nach Ablauf der Kündigungsandrohungsfrist ausgelöst werden). Es protokolliert nicht, wer den Übergang wann ausgelöst hat.

Das führt in der Praxis zu Problemen, die schwer zu erkennen sind, weil sie keine Fehlermeldung erzeugen. Ein Mieter bestreitet die Mahnung – die Verwaltung hat keinen lückenlosen Nachweis, dass die Mahnung tatsächlich versendet wurde, bevor der Status geändert wurde. Eine Mieterhöhung wird angefochten – die Verwaltung kann nicht nachweisen, ob das Zugangsdatum korrekt erfasst wurde. Eine Reparaturrechnung taucht auf – unklar, ob der Eigentümer die Beauftragung tatsächlich genehmigt hatte.

State Machines: der theoretische Unterbau

GenioFlow verwendet [XState v5](#), eine Bibliothek für endliche Automaten und Zustandsmaschinen in TypeScript. Die Entscheidung für einen bewährten Standard statt einer Eigenentwicklung hat einen praktischen Grund: Zustandsmaschinen sind formal verifizier-

bar. Eine State Machine kann in einem Diagramm dargestellt werden, das exakt zeigt, welche Übergänge möglich sind und welche nicht. Sie kann in automatisierten Tests gegen alle möglichen Eingaben getestet werden. Und sie trennt die Prozesslogik sauber von der Persistenzschicht.

In der Praxis sieht eine Zustandsmaschine für das Mahnwesen ungefähr so aus:

- Startzustand: `kein_rueckstand`
- Übergang `rueckstand_erkannt` → Zustand `rueckstand_erkannt`
- Übergang `erinnerung_senden` → Zustand `erinnerung`
- Übergang `mahnung_1_senden` → Zustand `mahnung_1`
- Übergang `mahnung_2_senden` → Zustand `mahnung_2`
- Übergang `mahnung_3_senden` → Zustand `mahnung_3` (Guard: Kündigungsandrohungsfrist abgelaufen)
- Übergang `inkasso_uebergeben` → Zustand `inkasso` (Terminal)
- Übergang `pay` → Zustand `bezahlt` (Terminal, aus allen aktiven Zuständen möglich)
- Übergang `stornieren` → Zustand `storniert` (Terminal, aus allen nicht-terminalen Zuständen möglich)

Die Guards sind der entscheidende Unterschied zu einem freien Statusfeld. Wenn der Guard `kuendigungsandrohungFristAbgelaufen` nicht erfüllt ist — weil das hinterlegte Fristdatum noch in der Zukunft liegt — wird der Übergang `mahnung_3_senden` von der Maschine verweigert. Nicht durch eine Validierungsregel im Frontend, sondern durch die Engine selbst, die serverseitig läuft.

Die drei aktuellen Workflow-Typen

Workflow Mahnwesen: 7 Stufen, automatische Zahlungserkennung

Das Mahnwesen ist der komplexeste der drei Workflow-Typen, weil es die meisten Zustände und die härtesten rechtlichen Anforderungen hat. Die Kündigungsandrohungsfrist aus § 543 Abs. 3 BGB — die Vorankündigung der außerordentlichen Kündigung bei erheblichem Zahlungsverzug — muss vor der dritten Mahnstufe abgelaufen sein. ImmoGenio erfasst das Fristende beim Versenden der zweiten Mahnung und sperrt den Übergang zur dritten Mahnstufe technisch, solange die Frist läuft.

Der Banking-Sync-Hook ist eine zweite wichtige Eigenschaft dieses Workflows. Nach jedem automatischen Kontoauszug-Import prüft ImmoGenio, ob offene Mahnungen vollständig beglichen wurden. Die Prüfung läuft gegen die `miet_zahlungen`-Tabelle: Wenn alle Positionen einer Mahnung vollständig bezahlt sind, löst der Hook automatisch den

`pay`-Übergang aus und schließt den Workflow ab. Der Verwalter muss nichts manuell nachhalten – der Mahnvorgang schließt sich von selbst, sobald die Zahlung eingegangen ist.

Der stündliche Cron-Job prüft zusätzlich alle aktiven Workflows auf Fälligkeit: Wenn eine Mahnstufe nach einer konfigurierten Wartezeit keine Reaktion erhalten hat, kann der Cron die nächste Stufe automatisch auslösen – sofern alle Guards erfüllt sind.

Workflow Mieterhöhung: § 558 BGB technisch erzwungen

Das Mieterhöhungsbegehren nach § 558 BGB ist in der Praxis fehleranfällig, weil es mehrere Fristen enthält. § 558b Abs. 2 BGB verlangt, dass zwischen dem Zugang des Erhöhungsverlangens und dem Beginn der Erhöhung mindestens drei Monate liegen. § 558a Abs. 1 BGB verlangt die Begründung des Verlangens – Mietspiegel, Gutachten oder Vergleichswohnungen. § 558 Abs. 3 BGB begrenzt die Erhöhung auf 20 % (Kappungsgrenze) innerhalb von drei Jahren.

ImmoGenio prüft alle drei Bedingungen serverseitig: Die Kappungsgrenze wird gegen die historischen Mietpreise berechnet, das Zugangsdatum wird beim Versenden erfasst, und der Übergang zur Umsetzung ist erst dann möglich, wenn der Fristablauf rechnerisch eingetreten ist. Eine Zustimmung des Mieters zu einem früheren Datum ist technisch erfassbar, ändert aber nichts an der gesetzlichen Mindestfrist – die Maschine unterscheidet zwischen dem Zustimmungsdatum und dem Gültigkeitsdatum der neuen Miete.

Der Workflow mündet entweder in `umgesetzt` – dann wird automatisch ein neuer Mietpreiseintrag in der Datenbank angelegt – oder in `zurueckgezogen`, wenn das Begehren zurückgezogen wird.

Workflow Instandsetzung: Eigentümerfreigabe ohne Portal-Account

Die Instandsetzung ist der Workflow-Typ mit der interessantesten technischen Eigenschaft: der öffentlichen Freigabe per Magic Link. Wenn die Hausverwaltung für eine Reparatur eine Eigentümerfreigabe einholen muss, generiert ImmoGenio serverseitig ein kryptografisch gesichertes Token (32 Byte aus dem Betriebssystem-CSPRNG, hex-kodiert). Dieses Token wird in den Workflow-Kontext geschrieben und per E-Mail an den Eigentümer versandt.

Der Freigabe-Endpunkt ist öffentlich erreichbar – er erfordert keine JWT-Authentifizierung. Das Sicherheitsmodell stützt sich stattdessen auf das Token selbst: 32 Byte entsprechen 2^{256} möglichen Werten, was Brute-Force praktisch ausschließt. Nach dem ersten Aufruf wird das Token als verbraucht markiert – Replay-Angriffe sind nicht möglich. Die Gültigkeit ist auf die konfigurierte Freigabefrist begrenzt.

Das Ergebnis: Ein Eigentümer kann eine Reparaturfreigabe erteilen oder ablehnen, ohne einen Portal-Account zu besitzen, ohne eine App zu installieren und ohne sich einzuloggen. Ein Klick in der E-Mail reicht.

Nebenläufigkeit und Skalierung: Was passiert bei gleichzeitigen Zugriffen?

Ein Detail, das in der Praxis selten diskutiert wird, ist die Nebenläufigkeit. Was passiert, wenn zwei Verwalter gleichzeitig versuchen, denselben Workflow-Übergang auszulösen? Was passiert, wenn der Cron-Job und ein manueller Klick zeitgleich eine Mahnstufe eskalieren wollen?

GenioFlow verwendet zwei komplementäre Sperrmechanismen. Die Engine selbst setzt innerhalb einer Datenbanktransaktion einen transaktionalen Advisory Lock (`pg_advisory_xact_lock`): Der Lock wird am Beginn der Transaktion erworben und mit dem Commit automatisch freigegeben. Zwei parallele Transition-Aufrufe für denselben Workflow können nicht gleichzeitig erfolgreich sein — einer wartet auf den anderen. Das ist eine Blocking-Strategie: kein Request geht verloren, aber einer von beiden muss warten.

Der Cron-Job verwendet eine andere Strategie: `pg_try_advisory_lock` auf Session-Ebene. Dieser Lock ist nicht blockierend: Wenn der Lock bereits gehalten wird (weil ein vorheriger Cron-Lauf noch läuft oder ein anderer Prozess denselben Workflow bearbeitet), überspringt der Cron diesen Workflow und macht mit dem nächsten weiter. Das verhindert, dass sich Cron-Läufe auftürmen und die Datenbank blockieren.

Der Lock-Namespace (`CRON_LOCK_NS = 2_346_000`) ist bewusst von den transaktionalen Locks der Engine getrennt, um Kollisionen zu vermeiden. Das ist ein Implementierungsdetail, das in der Benutzeroberfläche unsichtbar bleibt, aber darüber entscheidet, ob das System unter Last korrekt funktioniert.

Compliance-Aspekte: Was der Audit-Trail für den Verwaltungsalltag bedeutet

Jeder Zustandsübergang in GenioFlow erzeugt einen Eintrag in der Tabelle `workflow_logs`: Ausgangszustand, Zielzustand, Event-Name, Akteur (Benutzer-ID und Rolle oder `system`), Zeitstempel und optionaler Payload. Diese Einträge werden nur geschrieben, nie überschrieben. Auch Administratoren können sie nicht nachträglich anpassen — die Tabelle hat keine UPDATE-Berechtigung für Anwendungskonten.

Im Portal sehen Verwalter die vollständige Chronologie jedes Vorgangs: wann die Mahnung angelegt wurde, wann sie versendet wurde, wann die Zahlung eingegangen ist und wann der Vorgang abgeschlossen wurde. Im Streitfall ist das mehr als ein Log – es ist ein strukturierter Nachweis mit Zeitstempeln und Akteur-Zuordnung.

Ein konkretes Beispiel: Ein Mieter bestreitet, jemals eine zweite Mahnung erhalten zu haben. In einer klassischen Software hätte der Verwalter allenfalls ein gesendetes PDF – ohne Nachweis, ob der Status vor oder nach dem Versand gesetzt wurde. In GenioFlow hingegen zeigt das Protokoll: Zustand `mahnung_1` wurde am 14. März um 09:12 Uhr durch Benutzer-ID 47 (Rolle: `verwalter`) mit dem Event `mahnung_2_senden` verlassen. Die zugehörige E-Mail-ID ist im Payload gespeichert. Drei Minuten später bestätigte der E-Mail-Provider den Versand. Dieses Protokoll kann weder vom Verwalter noch vom Administrator nachträglich verändert werden – die Tabelle `workflow_logs` hat auf Datenbankebene keine UPDATE-Berechtigung für Anwendungskonten. Der Mieter kann den Erhalt bestreiten; dass die Mahnung korrekt aus dem Workflow heraus ausgelöst wurde, ist damit technisch dokumentiert.

Der Sync-Trigger (eine PostgreSQL `AFTER UPDATE`-Trigger-Funktion) schreibt den aktuellen Workflow-Zustand zusätzlich in die operativen Tabellen zurück: `mahnungen.status`, `mieterhoehungen.status`, `anfragen.status`. Das bedeutet: Alle bestehenden Status-Filter, Auswertungen und Exporte funktionieren unverändert – sie lesen aus den gleichen Tabellen wie zuvor, aber die Werte werden jetzt von der Workflow-Engine gesetzt.

Rollout ohne Risiko: Feature-Flag und Backfill

Für den produktiven Einsatz war ein sicherer Rollout-Pfad wichtig. GenioFlow wurde mit einem Feature-Flag (`GENIOFLOW_ENABLED`) ausgestattet: Mit dem Wert `false` arbeiten Mahnwesen, Mieterhöhung und Anfragen exakt wie vor der Einführung – alle Workflow-Aufrufe werden übersprungen. Ein Migrations-Rollback ist nicht nötig; bestehende Workflow-Daten bleiben erhalten und können jederzeit reaktiviert werden.

Die Backfill-Migration überträgt alle bestehenden Mahnungen, Mieterhöhungen und Anfragen nachträglich in die Workflow-Tabellen. Der aktuelle Zustand wird aus dem Datenbankstatus abgeleitet – eine Mahnung im Status `versandt` startet im Workflow-Zustand `mahnung_1`, `mahnung_2` oder `mahnung_3`, je nach Mahnstufe. Bestehende Vorgänge werden damit nahtlos in die neue Engine integriert, ohne dass die Nutzungserfahrung unterbrochen wird.

Die Kombination aus Feature-Flag und Backfill bedeutet: Ein Mandant kann GenioFlow in einer kontrollierten Testphase aktivieren, den Betrieb in der neuen Engine über mehrere Wochen beobachten und bei Bedarf – ohne Datenverlust – auf den Ausgangszustand zu-

rückschalten. Für Hausverwaltungen, die in der Vergangenheit schlechte Erfahrungen mit erzwungenen Software-Umstellungen gemacht haben, ist das ein erheblicher Unterschied: kein Stichtag, kein Alles-oder-Nichts.

Warum das der entscheidende Reifesprung ist

Buchungsregel-Engines (vgl. den Artikel zur automatischen Buchungszuordnung) sorgen dafür, dass Zahlungseingänge korrekt verbucht werden. OCR-Pipelines (vgl. [KI-Belegerfassung](#)) reduzieren den manuellen Erfassungsaufwand bei Eingangsberechnungen. Beide Systeme arbeiten reaktiv: Sie verarbeiten Daten, die bereits vorliegen.

GenioFlow ist anders. Es ist proaktiv: Es steuert, wie ein Prozess abläuft, bevor Fehler entstehen. Es verhindert nicht, dass ein Verwalter eine falsche Entscheidung trifft — das kann kein System leisten. Aber es verhindert, dass eine falsche Entscheidung ohne Spuren bleibt oder dass ein rechtlich unzulässiger Schritt technisch ausgeführt werden kann.

Für Hausverwaltungen, die professionell wachsen wollen, ist das die entscheidende Verschiebung: weg vom Werkzeug, das ausführt, was der Nutzer eingibt, hin zur Software, die den Prozess kennt und Grenzen setzt, wo der Gesetzgeber sie gesetzt hat.

GenioFlow ist Teil von ImmoGenio, der Verwaltungssoftware für moderne Hausverwaltungen. Die Workflow-Engine ist für alle Mandanten ohne zusätzliche Kosten verfügbar. Weitere Informationen unter </funktionen/genioflow/>.
