FLOWTRIX · BENUTZER- UND ADMINISTRATORHANDBUCH

Leistungsbeschreibung aus Fotos

Ein Foto in den Messenger — die KI liefert den fertigen Rechnungstext zurück.

Version	2.0
Stand	16.08.2026
Status	Einsatzbereit
System	n8n-Workflow · Telegram · WhatsApp · Threema · Vision-KI · Lexware (PUT-Weg deaktiviert)
Herausgeber	FlowTrix · flowtrix.de

Inhaltsverzeichnis

KAPITEL 1

Was ist das System?

Ein n8n-Workflow, der aus einem Baustellenfoto einen fertigen **Textvorschlag für die Rechnungsposition** macht. Der Monteur schickt ein Foto vom fertigen Auftrag über seinen Messenger — **Telegram, WhatsApp oder Threema**. Die KI beschreibt die sichtbare Leistung und schickt den Vorschlag **auf demselben Kanal zurück**, von dem das Foto kam. Der Nutzer übernimmt ihn per Copy-&-Paste in die Lexware-Rechnung.

> **Status: Einsatzbereit.** Drei Messenger als Eingang (Telegram, WhatsApp, Threema), Bildbeschreibung über eine Vision-KI (Cloud oder lokal umschaltbar), Antwort auf demselben Kanal zurück über den zentralen Messenger-Baustein.

> **Es wird nichts automatisch in Lexware geschrieben.** Der aktive Weg gibt nur einen Textvorschlag zurück. Der direkte Schreibweg (PUT) ist bewusst deaktiviert — siehe Kapitel 9.

KAPITEL 2

Voraussetzungen

- Laufende **n8n-Instanz**
- Mindestens einer der drei Messenger-Zugänge: **Telegram-Bot**, **WhatsApp Business Cloud** (Meta-Token) oder **Threema-Gateway** — mehrere parallel möglich
- Der zentrale **Messenger-Baustein** (Versand) mit den entsprechenden Zugängen — schickt die Antwort auf demselben Kanal zurück
- **Eine Vision-KI** (OpenAI-kompatibel) — **Cloud oder lokal** über `aiBaseUrl` / `aiModel` umschaltbar
- **Lexware** nur für den bewusst deaktivierten PUT-Weg (nicht für den Standardbetrieb nötig)

KAPITEL 3

Erste Schritte

1. Workflow-Datei in n8n importieren.
2. Mindestens einen Eingang einrichten: **Telegram** (Credential + Option „Download Image/File"), **WhatsApp** (Trigger-Credential; die HTTP-Nodes nutzen Header Auth mit dem Meta-Token) oder **Threema** (Callback-URL des Webhooks im Gateway).
3. Node **„Vision-Anfrage"** → Credential **HTTP Header Auth** mit `Authorization = Bearer <API-SCHLÜSSEL>` (oder `aiBaseUrl` auf ein lokales Modell zeigen).
4. Node **„Baustein: Antwort senden"** → den Sub-Workflow **„Messenger – Nachricht senden (Baustein)"** wählen.
5. In **Config** `aiBaseUrl` / `aiModel` prüfen; `binaerProperty` = `data`.
6. **Workflow aktiv schalten** — sonst empfängt kein Kanal Fotos.
7. Ein Foto (optional mit Bildunterschrift) an einen der Kanäle schicken und das Ergebnis prüfen.

> **Antwort auf demselben Kanal:** Antwort-Kanal und Antwort-Ziel werden bei der Normalisierung des jeweiligen Eingangs gesetzt — die Antwort geht automatisch an den Absender zurück, eine feste Chat-ID ist nicht nötig.

KAPITEL 4

Benutzer und Rollen

Rolle	Aufgabe	Braucht
Monteur (Baustelle)	Foto vom fertigen Auftrag über seinen Messenger schicken	Telegram / WhatsApp / Threema
Nutzer (Büro)	übernimmt den Vorschlag in die Lexware-Rechnung	Lexware
Administrator	richtet Messenger-Zugänge, KI-Credential, Messenger-Baustein und Config ein	n8n-Zugang

KAPITEL 5

Hauptfunktionen (aktive Kette)

5.1 Foto-Empfang über drei Messenger

Drei Trigger nehmen das Foto entgegen: **Telegram** (Option „Download Image/File"), **WhatsApp** (Graph-API holt über die Medien-ID die Bilddatei) und **Threema** (Gateway-Webhook). Eine optionale Bildunterschrift dient als Zusatzkontext.

5.2 Normalisieren

Jeder Eingang wird auf eine einheitliche Form gebracht (Foto-Binär → `data`, plus Antwort-Kanal und Antwort-Ziel). Danach laufen alle drei Kanäle durch denselben Ablauf.

5.3 Bild vorbereiten

Das Foto-Binär wird in eine **base64-data-URL** verpackt. Daraus entstehen die Vision-`messages`: ein System-Prompt und der User-Inhalt mit Text und `image_url`.

5.4 Vision-Anfrage

POST an `{{aiBaseUrl}}/chat/completions` mit `model` = `aiModel` — **Cloud oder lokal**. Der System-Prompt bittet um eine knappe, sachliche Rechnungsposition — kein Marketing, keine Preis- oder Mengenvermutungen.

5.5 Vorschlag bauen & Antwort auf demselben Kanal

Aus der KI-Antwort entstehen der Nachrichtentext und eine `lineItem`-Vorlage für Lexware (Menge/Preis bleiben 0). Der zentrale **Messenger-Baustein** schickt den Text an den Absender zurück — über genau den Kanal, von dem das Foto kam.

5.6 Foto-Wächter

Nachrichten ohne verwertbares Foto-Binär werden erkannt und übersprungen. Nichts wird in Lexware geschrieben — das Übernehmen bleibt beim Menschen.

KAPITEL 6

Typischer Arbeitsablauf

1. Monteur schickt ein Foto vom fertigen Auftrag über Telegram, WhatsApp oder Threema (optional mit Bildunterschrift).
2. Der Workflow normalisiert den Eingang, verpackt das Foto und schickt es an die Vision-KI.
3. Die KI beschreibt die sichtbare Leistung als Rechnungsposition.
4. Der Vorschlag kommt über den Messenger-Baustein **auf demselben Kanal zurück**; der Nutzer übernimmt ihn in die Lexware-Rechnung und ergänzt Menge und Preis.

KAPITEL 7

Die Ansicht

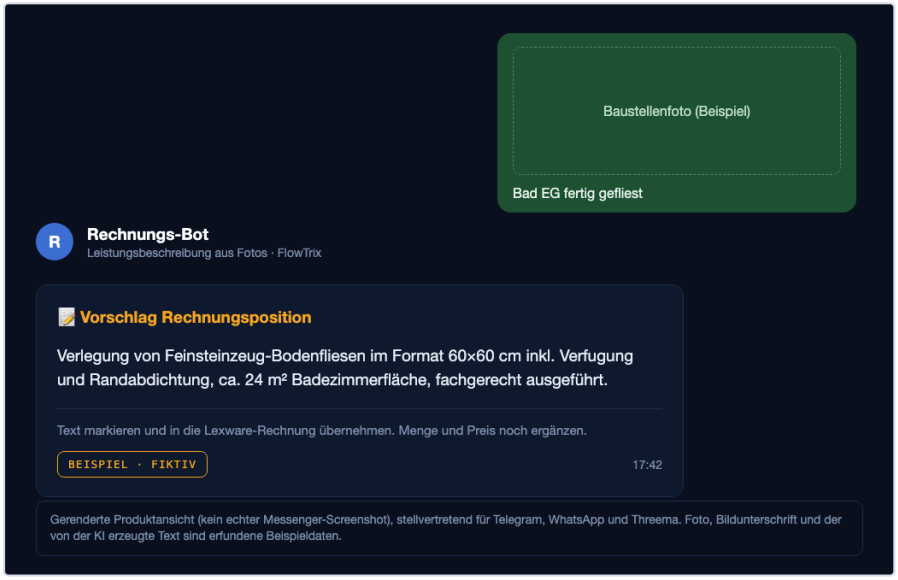

Abbildung 1: Vorschlag des Rechnungs-Bots im Chat — gerenderte Produktansicht, stellvertretend für alle drei Kanäle. Foto, Bildunterschrift und Text sind Beispieldaten (fiktiv).

KAPITEL 8

Administration

Alle Einstellungen liegen im Node **Config**.

Feld	Bedeutung	Standard
`aiBaseUrl`	OpenAI-kompatible Basis (inkl. `/v1`), Cloud oder lokal	`…openai.com/v1`
`aiModel`	Vision-Modell	`gpt-4o-mini`
`binaerProperty`	Name des Foto-Binärfeldes (nach Normalisierung einheitlich)	`data`
`rechnungId`	nur für den deaktivierten PUT-Weg	`<ID>`

Antwort-Kanal und Antwort-Ziel kommen aus der Normalisierung — die Antwort geht an den Absender zurück, eine feste Chat-ID ist nicht mehr nötig.

Zugangsdaten

Nur Platzhalter — echte Werte liegen ausschließlich im n8n-Credential-Store.

Dienst	Credential
Telegram	Bot-Token
WhatsApp Business Cloud	Trigger-Credential + HTTP Header Auth (Meta-Token) für die Graph-API
Threema	Gateway-Zugang (Webhook-Callback)
Vision-KI	HTTP Header Auth · `Authorization = Bearer <API-SCHLÜSSEL>` (bei lokalem Modell entfällt der Key)

KAPITEL 9

Rechnung schreiben — bewusst nicht aktiv

Die Position **direkt** an eine Lexware-Rechnung anzuhängen ist heikel: die API kennt nur **Full-Replacement per PUT** (`GET /invoices/{id}` → komplettes Objekt inkl. `version` zurückschreiben). Ein Fehler überschreibt die ganze Rechnung.

> **Deaktivierte Node-Insel.** Der PUT-Weg ist ohne Verbindung zur aktiven Kette vorbereitet. Zum Aktivieren: Insel an „Vorschlag bauen" hängen, die vier Nodes aktivieren, Baustein wählen, `rechnungId` setzen — nur bei **DRAFT-Rechnungen**, nach Freigabe im Betrieb.

KAPITEL 10

Datenschutz und Sicherheit

- Verarbeitet werden **Foto, Bildunterschrift und der erzeugte Beschreibungstext**. Fotos können personenbezogene Inhalte zeigen.
- Empfänger: die eigene n8n-Instanz, der genutzte Messenger (Telegram/WhatsApp/Threema) und der gewählte KI-Dienst.
- Der aktive Weg schreibt nichts in Lexware — das Übernehmen bleibt beim Menschen. Die Antwort geht ausschließlich an den Absender zurück.
- Zugangsdaten liegen ausschließlich im n8n-Credential-Store.
- **Zu bedenken:** Foto und Text laufen über einen KI-Dienst — das gehört in die Datenschutz-Betrachtung des Betriebs. Für datensparsamen Betrieb lässt sich über `aiBaseUrl` ein lokales Vision-Modell einsetzen. Der Zugang kann auf bekannte Absender beschränkt werden.

KAPITEL 11

Fehlerbehebung

Problem	Mögliche Ursache	Lösung
KI bekommt kein Bild / leere Beschreibung	Base64 nicht im Binär-Feld oder falscher Feldname	`binaerProperty` prüfen (nach Normalisierung `data`); sonst Foto über eine öffentliche URL übergeben
Kanal reagiert nicht	Workflow nicht aktiv / Telegram-„Download" aus / Threema-Callback-URL fehlt	aktiv schalten; Trigger-Option bzw. Webhook-URL im Gateway setzen
Antwort kommt nicht zurück	Messenger-Baustein nicht gewählt / Zugang im Baustein fehlt	„Baustein: Antwort senden" auf den Versand-Sub-Workflow setzen, Kanal-Zugang prüfen
401 bei Vision-Anfrage	Header-Auth-Key fehlt oder falsch	Bearer-Key im Credential prüfen
WhatsApp lädt kein Bild	Meta-Token fehlt/abgelaufen	Header-Auth-Token für die Graph-API prüfen

KAPITEL 12

Häufige Fragen & Einschränkungen

Muss der Monteur etwas installieren?
Nein. Er schickt nur ein Foto über den Messenger, den er ohnehin nutzt — Telegram, WhatsApp oder Threema.

Kommt die Antwort auf demselben Kanal zurück?
Ja. Wer per WhatsApp schickt, bekommt die Antwort per WhatsApp; dasselbe gilt für Telegram und Threema. Das übernimmt der zentrale Messenger-Baustein.

Welche KI kann ich nutzen?
Eine OpenAI-kompatible Vision-KI. Über `aiBaseUrl` / `aiModel` zwischen Cloud und lokalem Modell (z. B. Ollama llava) umschaltbar.

Schreibt das direkt in Lexware?
Nein. Der aktive Weg gibt nur einen Textvorschlag zurück; der PUT-Weg ist bewusst deaktiviert.

Bekannte Einschränkungen

- Menge und Preis der Rechnungsposition ergänzt der Nutzer (bewusst — keine Preisvermutung durch die KI).
- Threema-Empfang setzt ein eingerichtetes Gateway voraus (Callback-URL, E2E-Entschlüsselung im Gateway).
- Der direkte Schreibweg (PUT) ist bewusst deaktiviert und nur für DRAFT-Rechnungen nach Freigabe gedacht.

KAPITEL 13

Support und Versionshistorie

FlowTrix · Kontakt: **https://flowtrix.de/kontakt/** · Projektübersicht: https://flowtrix.de/projekte/

Version	Datum	Änderung
2.0	16.08.2026	Mehrkanal-Version: drei Messenger als Eingang (Telegram, WhatsApp, Threema), Normalisierung, zentraler Messenger-Baustein für den Versand (Antwort auf demselben Kanal zurück), Vision-KI Cloud oder lokal umschaltbar. Einsatzbereit.
1.0	22.07.2026	Erstausgabe. Foto-Empfang (Telegram), Bild vorbereiten, Vision-Anfrage, Vorschlag bauen, Telegram-Rückgabe. PUT-Weg deaktiviert.