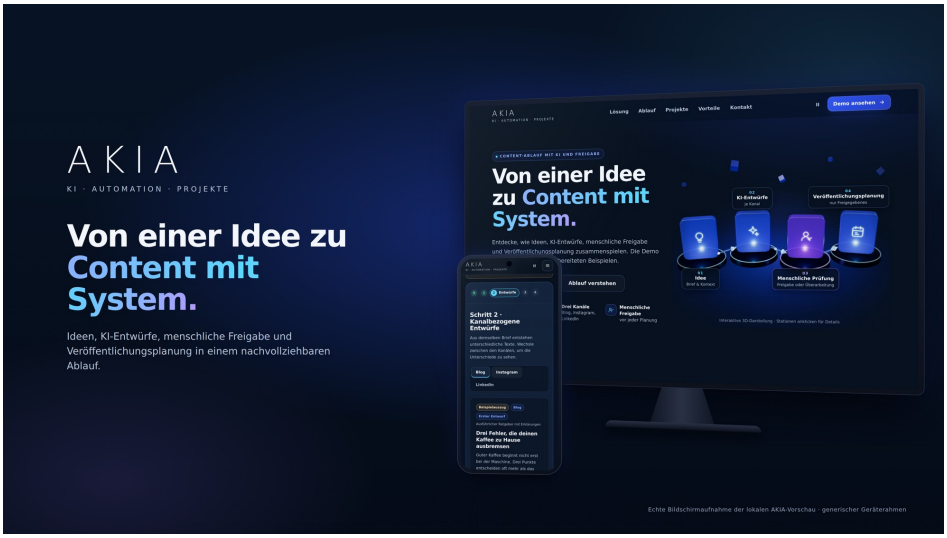
AKIA · KI – Automation – Projekte

Projektaufgabe 2

Automatisiertes Content-System mit Human-in-the-Loop

Konzeption, Umsetzung, Test und Deployment mit n8n, PostgreSQL, OpenAI, NocoDB und ergänzenden Cloud-Diensten



Projektmotiv der öffentlichen Demo

Projektdokumentation	Projektaufgabe 2 (PA2)
Verfasser	Alexander Ziersch
Abgabestand	05. 10. 2026
Systemstand	finaler Live-Test 05. 10. 2026
Öffentliche Demo	https://www.akia-projekte.de/projektaufgabe2/

Kurzfassung

PA2 ist die technische und prozessuale Weiterentwicklung von Projektaufgabe 1 (PA1). Als konsistenter Demonstrationskontext wurde erneut die vollständig fiktive Kaffeerösterei „Kontor 12“ verwendet. Während PA1 einen einzelnen KI-gestützten Instagram-Post demonstrierte, erweitert PA2 diesen Ansatz zu einem modularen Mehrkanal-Content-System mit strukturierter Datenhaltung, Human-in-the-Loop, Bildgenerierung, Live-Publishing und Monitoring. Kontor 12 dient ausschließlich als fiktiver Test- und Demonstrationsfall.

Ziel der Projektaufgabe war ein nachvollziehbares, automatisiertes Content-System, das aus einer Idee kanalbezogene Inhalte erzeugt, diese prüft und versioniert und sie erst nach menschlicher Freigabe an die Publishing-Vorbereitung übergibt. Die Lösung ist nicht als monolithischer n8n-Workflow umgesetzt, sondern als vier fachliche Kernworkflows plus einen separat veröffentlichten Prüfer-Demo-Orchestrator.

> **Ergebnis im Überblick**
>
> Ein zeitgesteuerter Workflow erzeugt KI-gestützt Content-Ideen. Ein Produktionsworkflow erstellt daraus Texte für Blog, Instagram und LinkedIn. Nach der Freigabe aller drei Kanäle erzeugt der Review-/Publishing-Workflow das Instagram-Bild, lädt es zu Cloudinary, veröffentlicht den Beitrag über die Buffer API live auf Instagram, protokolliert den Publikationsstatus und versendet eine Benachrichtigung. PostgreSQL dient als zentrale Datenbasis; ein vierter Kernworkflow übernimmt die Fehlerbehandlung. Zusätzlich orchestriert `PA2_Demo_Instagram` den öffentlich auslösbaren Prüfer-End-to-End-Test, ohne die fachlichen Kernworkflows zu duplizieren.

Abgrenzung. Die direkte Instagram-Veröffentlichung ist im dokumentierten Systemstand aktiv und wurde über Buffer erfolgreich live getestet. Das generierte Bild wird zuvor über Cloudinary öffentlich erreichbar gemacht. Die Meta-Developer-Anbindung bleibt als alternative technische Variante dokumentiert, ist für den produktiven Publishing-Pfad jedoch nicht erforderlich.

Öffentliche Testzugänge

Auswahlseite	https://www.akia-projekte.de/projektaufgabe2/
Klassisch	https://www.akia-projekte.de/projektaufgabe2/klassisch/
3D	https://www.akia-projekte.de/projektaufgabe2/3d/
Cyberflug	https://www.akia-projekte.de/projektaufgabe2/cyberflug/
Live-Demo	https://www.akia-projekte.de/projektaufgabe2/demo-instagram/

Tabelle 1: Umsetzungsstand der Systembausteine (Systemstand 05. 10. 2026)

Baustein	Umsetzung	Status
Automatische Ideen	Schedule, KI-Idee, PostgreSQL, Production-Webhook	umgesetzt und getestet
Content-Produktion	Blog, Instagram, LinkedIn; Validierung; Versionierung	umgesetzt und getestet
Human-in-the-Loop-Review	drei Freigaben je Job, PENDING → APPROVED in NocoDB	Freigabepfad umgesetzt und getestet; Ablehnungspfad ohne Überarbeitungszyklus
Publishing	Publish-Payload, Bild, Cloudinary, Buffer, Instagram, Publikationsdatensatz	umgesetzt und live getestet
Benachrichtigung	Gmail API über n8n	umgesetzt und getestet
Error-Handling	zentraler Error Trigger, Protokollierung, Klassifizierung, Final-Fail-Pfad	umgesetzt; Retry-Härtung empfohlen
Instagram-Live-Publishing	Buffer GraphQL API mit öffentlicher Cloudinary-Bild-URL	umgesetzt und live getestet
Öffentliche Projektdemo	Ansichten „Klassisch“, „3D“ und „Cyberflug“ auf Hetzner/nginx	online
Prüfer-End-to-End-Demo	Website-Webhook → Demo-Idee → Produktion → Demo-Freigabe → Review/Publishing → Instagram → Response	umgesetzt und live getestet

Inhaltsverzeichnis

Kurzfassung i

Abbildungsverzeichnis v

Tabellenverzeichnis vi

Abkürzungsverzeichnis vii

1 Ausgangssituation und Zielsetzung 1
- 1.1 Projektziele . . . 1
- 1.2 Begriffsfestlegung und Bezugszeitpunkt . . . 2

2 Lösungsarchitektur 3

3 Technische Umgebung und Datenmodell 5
- 3.1 Zentrale Tabellen . . . 5
- 3.2 Statusmodell . . . 5

4 Workflow A – Automatische Ideen 7

5 Workflow B – Content-Produktion 9
- 5.1 Verarbeitungsschritte . . . 11
- 5.2 Konsistenz und Validierung . . . 11

6 Human-in-the-Loop-Review 12

7 Workflow C – Review und Publishing 13

8 Zentrales Error-Handling 17
- 8.1 Umgesetzter Funktionsumfang . . . 17

9 Teststrategie und Nachweise 19
- 9.1 Negativtests . . . 19
- 9.2 Beispiel: automatische Ende-zu-Ende-Kette . . . 19

10 Öffentliche Website und Deployment 22
- 10.1 Deployment-Nachweis . . . 24

11 Sicherheits- und Betriebsaspekte 25

12 Offene Punkte und Weiterentwicklung 26

13 Fazit 27

Quellenverzeichnis **28**

A Kernstatus und Prüfcheckliste **29**

B Modelle und Prompt-Vorlagen **30**

Abbildungsverzeichnis

2.1 Logische Gesamtarchitektur des PA2-Systems 3
2.2 Veröffentlichtes Workflow-Set in n8n 4

3.1 Statusmodell der statusführenden Entitäten 6

4.1 Die KI-Idee wird als QUEUED-Datensatz in PostgreSQL gespeichert 7
4.2 Der Production-Webhook empfängt die übergebene Ideen-ID 8

5.1 Workflow „PA2-Produktion“ vom Webhook bzw. manuellen Start bis zu den Abschlussdaten 10

6.1 NocoDB-Ansicht mit Freigaben in den Status APPROVED und PENDING 12

7.1 Finaler Review- und Publishing-Workflow mit Live-Veröffentlichung über Buffer . 14
7.2 Der Cloudinary-Upload liefert eine öffentlich erreichbare URL 16
7.3 Empfangene n8n-Statusmail; der gezeigte Nachweis stammt aus einem früheren Publishing-Test 16

8.1 Zentraler Error-Workflow mit Retry-Prüfung und Final-Fail-Pfad 17

10.1 Klassische Projektansicht der AKIA-Demo 22
10.2 Erfolgreicher Prüfer-End-to-End-Lauf: Produktion und Publishing werden über den Demo-Orchestrator bis zur Response gekoppelt 23
10.3 Responsive Darstellung der Projektseite 24

Tabellenverzeichnis

1 Umsetzungsstand der Systembausteine (Systemstand 05. 10. 2026) ii

1.1 Begriffsfestlegung 2

2.1 Kopplungsprinzipien zwischen den Workflows 4

3.1 Systemkomponenten und ihre Rolle 5
3.2 Zentrale Tabellen im Schema `content` 6

4.1 Nodes von Workflow A in Ausführungsreihenfolge 7

5.1 Verarbeitungsphasen von Workflow B 11

6.1 Freigabezustände und ihre Folgen 12

7.1 Nodes von Workflow C in Ausführungsreihenfolge 15

9.1 Durchgeführte Testfälle 20
9.2 Negativtests 21

10.1 Bereitgestellte Ansichten der Projektdemo 23

11.1 Sicherheits- und Betriebsaspekte 25

12.1 Offene Punkte nach Priorität 26

13.1 Zielerreichung 27

A.1 Prüfcheckliste zum Systemstand 29

B.1 Eingesetzte Modelle und Parameter 30

Abkürzungsverzeichnis

API	Application Programming Interface
HITL	Human-in-the-Loop
HTTP(S)	Hypertext Transfer Protocol (Secure)
ID	Identifikator
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
PA1/PA2	Projektaufgabe 1 bzw. 2
SQL	Structured Query Language
SSH	Secure Shell
TLS	Transport Layer Security
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

1 Ausgangssituation und Zielsetzung

Ausgangspunkt war ein Content-Prozess mit zahlreichen manuellen Einzelschritten: Themenidee formulieren, Kontext zusammenstellen, Inhalte je Kanal erstellen, Qualität prüfen, Freigaben einholen und Ergebnisse dokumentieren. Zwischenstände und Freigaben werden in einem solchen Ablauf nicht systematisch festgehalten und sind daher schwer nachvollziehbar. PA2 ersetzt diese lose Kette durch ein orchestriertes System mit definierten Statusübergängen und nachvollziehbarer Datenhaltung.

1.1 Projektziele

Z1 Eine Idee liegt manuell oder automatisch erzeugt als strukturierter Datensatz in PostgreSQL vor.

Z2 Für Blog, Instagram und LinkedIn entstehen getrennte KI-Texte auf Basis eines gemeinsamen Produktionsbriefs.

Z3 Die Ergebnisse werden technisch validiert, versioniert und je Kanal als Freigabeobjekt gespeichert.

Z4 Ohne menschliche Freigabe läuft der Publishing-Pfad nicht weiter.

Z5 Freigegebene Inhalte werden automatisch für die Veröffentlichung vorbereitet, mit einem Bild ergänzt und nachvollziehbar protokolliert.

Z6 Fehler werden zentral erfasst, und der Jobstatus wird nachvollziehbar aktualisiert.

Z7 Das System wird als öffentliche Demo dokumentiert, ohne interne Zugangsdaten oder Backend-Zugriffe offenzulegen.

> **Architekturentscheidung**
>
> Die menschliche Freigabe kann Minuten, Stunden oder Tage dauern. Deshalb ist der Prozess in mehrere Workflows aufgeteilt. Produktion und Publishing-Vorbereitung sind über den in PostgreSQL geführten Status gekoppelt; eine dauerhaft wartende n8n-Ausführung ist nicht erforderlich.

1.2 Begriffsfestlegung und Bezugszeitpunkt

Um Mehrdeutigkeiten zu vermeiden, werden die folgenden Begriffe im gesamten Dokument einheitlich verwendet. Alle Statusangaben beziehen sich auf den Systemstand vom 05. 10. 2026.

Tabelle 1.1: Begriffsfestlegung

Begriff	Bedeutung in diesem Dokument
Publishing	Erzeugung von Publish-Payload und Bild, Ablage bei Cloudinary, Live-Veröffentlichung über Buffer sowie Anlage bzw. Aktualisierung des Publikationsdatensatzes (Workflow C).
Veröffentlichung	Tatsächliche Publikation auf Instagram über die Buffer GraphQL API; bei erfolgreichem Live-Post wird der Status PUBLISHED gesetzt.
Freigabe (Approval)	Menschliche Prüfentscheidung je Kanal in `content.approvals`.
Production-Webhook	Produktions-URL des Webhook-Triggers von Workflow B; einheitlich so bezeichnet.
Job	Ein Produktionsauftrag in `content.production_jobs`, der genau eine Idee für drei Kanäle verarbeitet.
Node	Einzelner Verarbeitungsschritt in einem n8n-Workflow. Die numerischen Präfixe sind workflowübergreifend eindeutig vergeben und bezeichnen Nodes, nicht die Ausführungsreihenfolge.

2 Lösungsarchitektur

Die fachliche Lösung besteht aus vier veröffentlichten Kernworkflows. Die drei Hauptprozesse A bis C bilden eine zusammenhängende Verarbeitungskette; Workflow D übernimmt die zentrale Fehlerbehandlung (Abbildung 2.1). Zusätzlich ist der veröffentlichte Orchestrierungsworkflow `PA2_Demo_Instagram` ausschließlich für den öffentlichen Prüfer-End-to-End-Test vorgesehen. Er stößt die vorhandenen Kernworkflows an, statt deren Logik zu duplizieren.

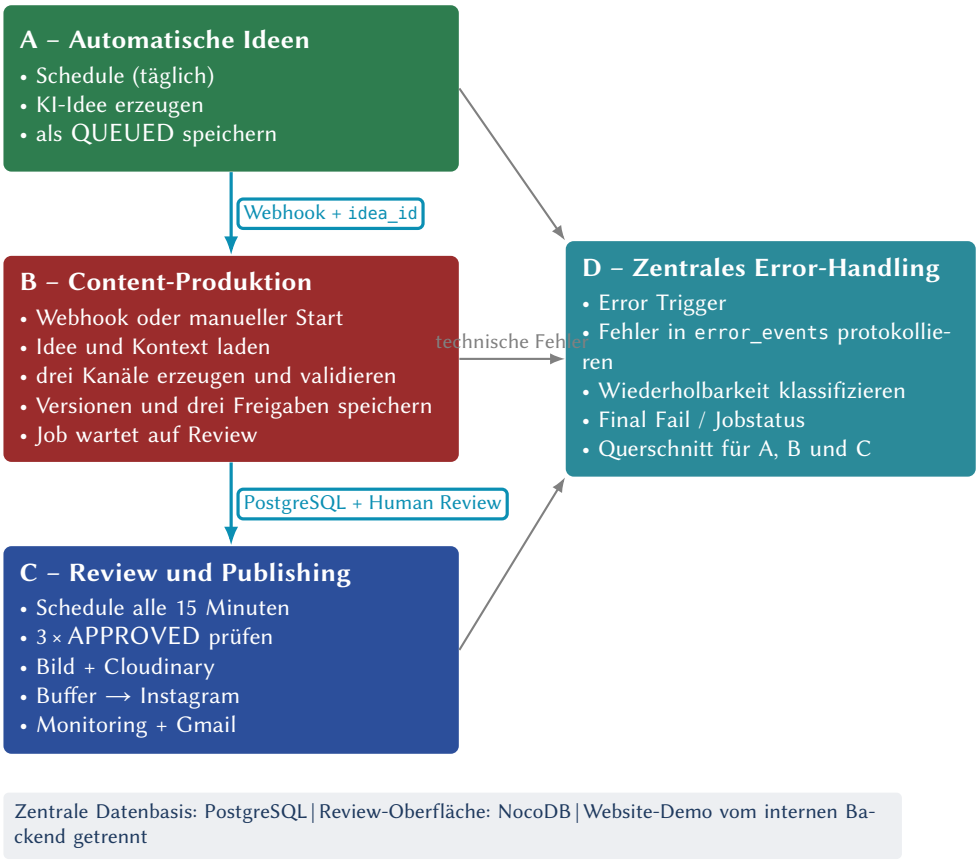


Abbildung 2.1: Logische Gesamtarchitektur des PA2-Systems (neu als Vektorgrafik gesetzt)

Tabelle 2.1: Kopplungsprinzipien zwischen den Workflows

Übergang	Mechanismus	Begründung
A ⟶ B	HTTP-POST auf den Production-Webhook mit `idea_id`	Gezielte Übergabe der gerade erzeugten Idee; die Produktion kann zusätzlich manuell gestartet werden.
B ⟶ C	PostgreSQL-Status + Human Review	Die Publishing-Vorbereitung startet asynchron erst nach 3 × APPROVED; es gibt keine lang wartende n8n-Ausführung.
A/B/C ⟶ D	n8n-Error-Workflow	Die Fehlerbehandlung bleibt als Querschnittsfunktion vom Normalpfad getrennt.

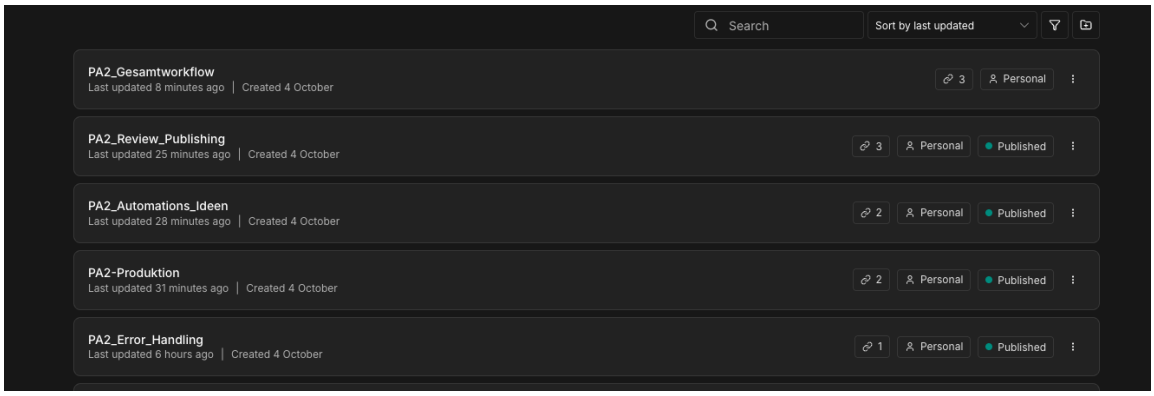


Abbildung 2.2: Veröffentlichtes Workflow-Set in n8n

3 Technische Umgebung und Datenmodell

Die Lösung wird selbst betrieben auf einem Hetzner-Server unter Ubuntu. nginx läuft als Systemdienst direkt auf dem Host und liefert die öffentliche Projektwebsite aus. n8n, PostgreSQL, NocoDB, Redis, Headscale sowie die n8n-Runner laufen containerisiert.

Tabelle 3.1: Systemkomponenten und ihre Rolle

Komponente	Rolle im System
Hetzner / Ubuntu	Hosting des gesamten Projektsystems
nginx	HTTPS und öffentliche Auslieferung von `akia-projekte.de`
n8n	Orchestrierung der Workflows und API-Aufrufe
PostgreSQL 17	Zentrale Datenbasis und Statusführung [1]
NocoDB	Review-Oberfläche für Freigaben und Dateninspektion
Redis	Infrastrukturkomponente des Self-Hosted-Stacks; nicht Bestandteil der fachlichen PA2-Logik
OpenAI	Textgenerierung und Bildgenerierung (GPT Image)
Cloudinary	Öffentlich erreichbarer Speicher für generierte Bilder
Buffer API	GraphQL-Schnittstelle für Queueing und Live-Publishing auf dem verbundenen Instagram-Kanal [2, 3]
Gmail API	Statusbenachrichtigung nach dem Publishing-Lauf
Headscale/Tailscale	Administrativer Zugriff über ein privates Overlay-Netz

3.1 Zentrale Tabellen

3.2 Statusmodell

Abbildung 3.1 fasst die Statusübergänge der statusführenden Entitäten zusammen. Im normalen Prozess endet die Produktion vor der menschlichen Freigabe; der Review-/Publishing-Pfad verarbeitet vollständig freigegebene Jobs im Status COMPLETED. Für den dedizierten Prüfer-Demo-Pfad setzt `PA2_Demo_Instagram` nach erfolgreicher Produktion ausschließlich den zugehörigen Demo-Job und seine drei Freigaben gezielt auf den für Workflow C erforderlichen Zustand. Der Ideenstatus dient primär der Reservierung und Produktionssteuerung; ein zusätzlicher fachlicher Archiv-Endstatus der Idee ist für den demonstrierten Kernprozess nicht erforderlich.

Tabelle 3.2: Zentrale Tabellen im Schema `content`

Tabelle	Zweck	Wichtige Felder / Status
`idea_sources`	Quellen für Content-Ideen	`source_type`, `automatic`, `active`
`content_ideas`	Ideen-Warteschlange	`source_id`, `title`, `priority`, `status`
`companies`	Firmen- und Markenkontext	Firmen- und Kontextdaten
`channels`	Kanaldefinitionen	`blog`, `instagram`, `linkedin`
`production_jobs`	Produktionsauftrag	QUEUED, RUNNING, WAITING_FOR_REVIEW, COMPLETED, FAILED
`production_job_channels`	Kanäle je Job	QUEUED, PROCESSING, COMPLETED, FAILED
`content_versions`	Versionierte Content-Ausgaben	`job_id`, `channel_id`, `version_number`; DRAFT
`approvals`	Menschliche Freigabe	PENDING, APPROVED, REJECTED; `reviewer`, `approved_at`
`publications`	Publikationsstatus	PENDING, SCHEDULED, PUBLISHING, PUBLISHED, FAILED, CANCELLED
`error_events`	Zentrales Fehlerprotokoll	`error_type`, `retryable`, `retry_count`, `resolved`

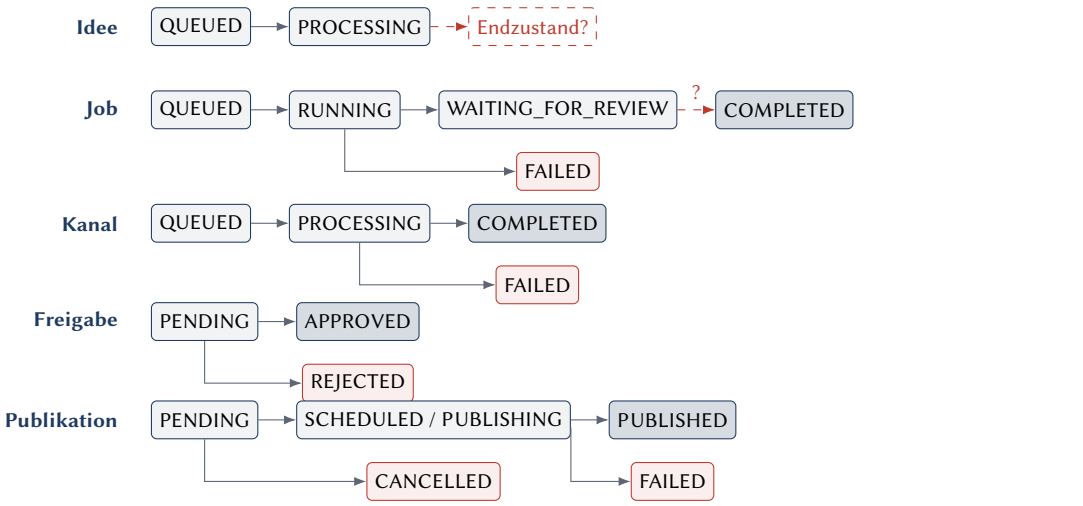


Abbildung 3.1: Statusmodell der statusführenden Entitäten

4 | Workflow A – Automatische Ideen

Der Workflow `PA2_Automations_Ideen` ergänzt den manuellen Ideenprozess um einen täglich ausgeführten Automationspfad. Ein Sprachmodell erzeugt eine strukturierte Content-Idee, die in PostgreSQL gespeichert und einer Quelle vom Typ `AI_GENERATED` zugeordnet wird. Anschließend ruft ein HTTP-Request den Production-Webhook des Workflows `PA2-Produktion` auf. Die Node-Präfixe sind workflowübergreifend vergeben (Tabelle 1.1); Tabelle 4.1 listet die Nodes in Ausführungsreihenfolge.

Tabelle 4.1: Nodes von Workflow A in Ausführungsreihenfolge

Node	Funktion
`22_Automatischer_Trigger`	Zeitgesteuerter Start, täglich
`21_Automatische_Ideengenerierung`	OpenAI erzeugt ein JSON-Objekt mit `title`, `idea_text`, `keywords`, `content_goal` und `priority`.
`21b_Idee_speichern`	`INSERT` in `content.content_ideas` mit Status QUEUED
`21c_Produktionsworkflow_starten`	POST an den Production-Webhook mit `idea_id`, `company_id` und `source_id`

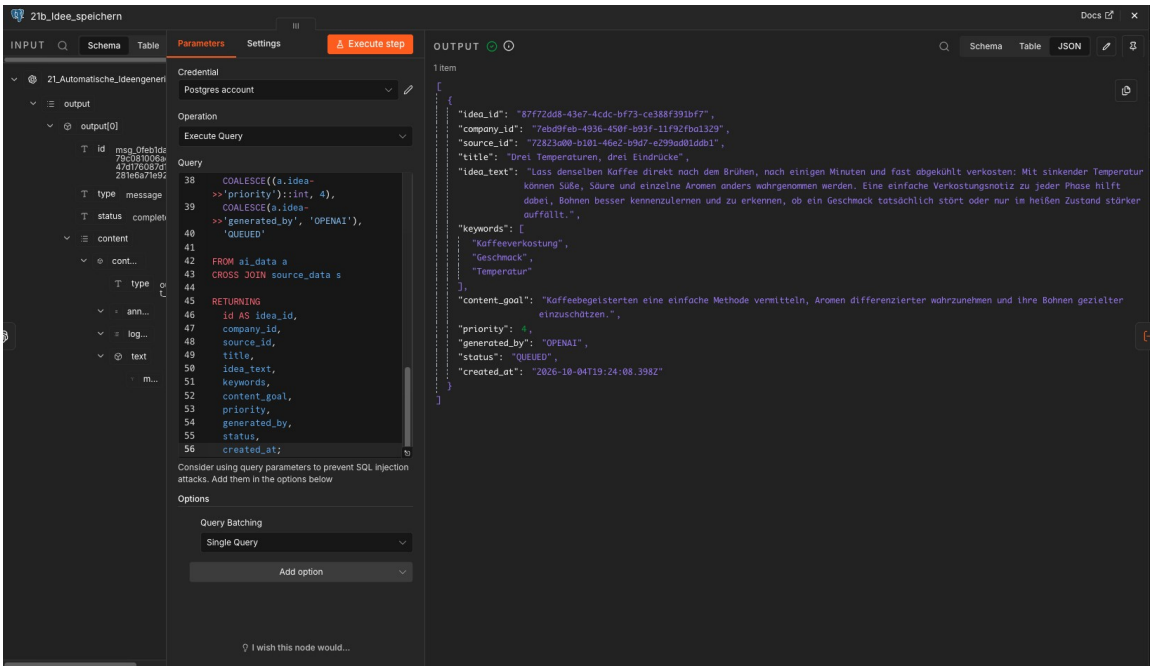


Abbildung 4.1: Die KI-Idee wird als QUEUED-Datensatz in PostgreSQL gespeichert

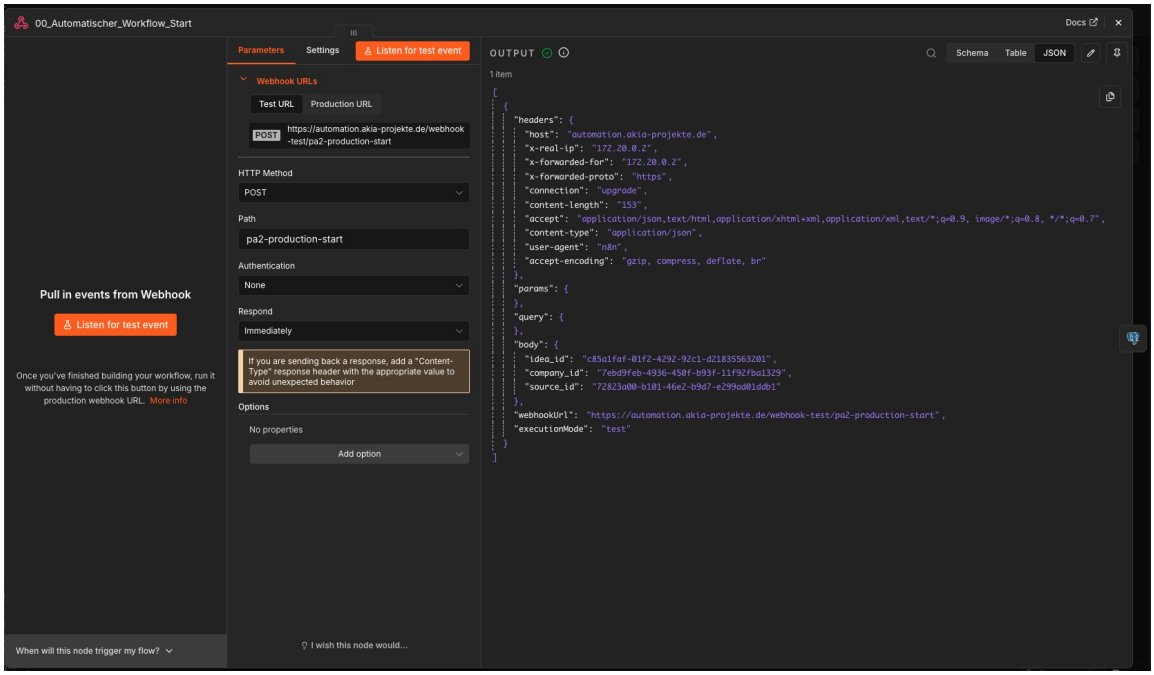


Abbildung 4.2: Der Production-Webhook empfängt die übergebene Ideen-ID

Gezielte Verarbeitung

Node 02 im Produktionsworkflow wertet `body.idea_id` aus. Bei automatischem Start wird genau die zuvor erzeugte KI-Idee geladen; bei manuellem Start wird weiterhin die nächste freie Idee im Status QUEUED gewählt.

5 | Workflow B – Content-Produktion

`PA2-Produktion` ist der fachliche Kern des Systems. Der Workflow besitzt zwei Einstiege, einen manuellen Start und den Production-Webhook. Danach wird genau eine Idee geladen, reserviert und als Produktionsjob für drei Kanäle verarbeitet. Abbildung 5.1 zeigt den Workflow im Überblick; maßgeblich für die Einzelschritte ist Tabelle 5.1.

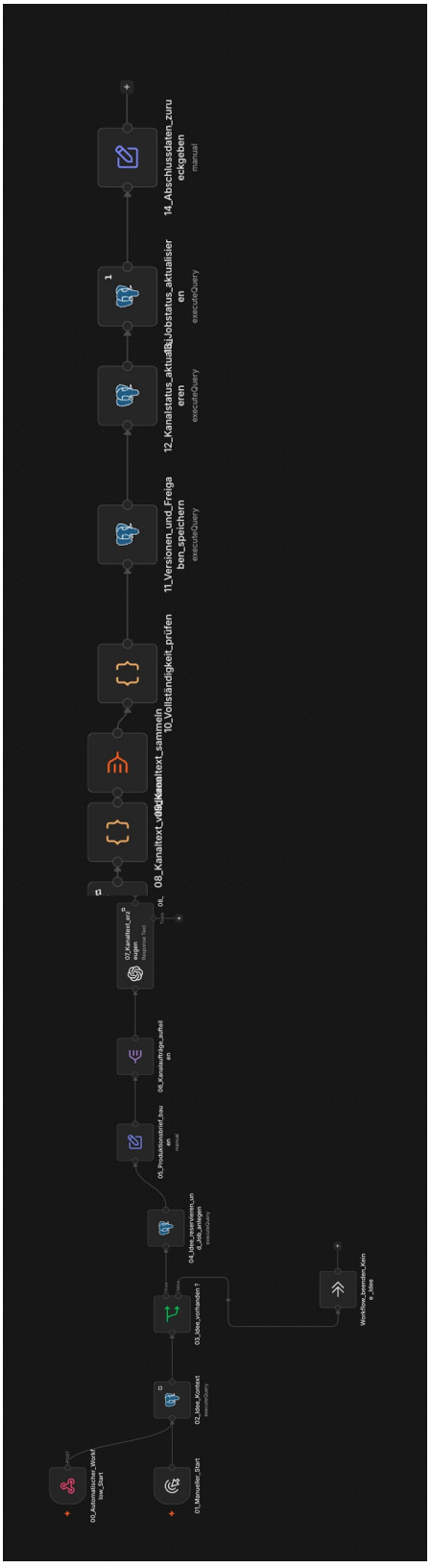


Abbildung 5.1: Workflow „PA2-Produktion“ vom Webhook bzw. manuellen Start bis zu den Abschlussdaten

5.1 Verarbeitungsschritte

Tabelle 5.1: Verarbeitungsphasen von Workflow B

Phase	Nodes	Ergebnis
Idee und Kontext	02–04	Idee laden, Existenz prüfen, Idee reservieren und Job anlegen
Produktionsbrief	05	Gemeinsamer Brief mit Ziel, Firma, Thema und Kanalvorgaben
Kanalaufteilung	06	Drei Items für Blog, Instagram und LinkedIn
KI-Erzeugung	07	Kanalbezogene Texte aus einem gemeinsamen Kontext
Längenabsicherung	07b	Technische Absicherung des Content-Body auf den Zielbereich; verhindert zu kurze/zu lange KI-Ausgaben
Validierung	08	Itemweise Prüfung von Pflichtfeldern, JSON-Struktur und Zeichenlänge (zulässig 300--1200 Zeichen)
Zusammenführung	09–10	Drei validierte Ergebnisse sammeln, auf Anzahl prüfen und Kanal-IDs für die Persistierung normalisieren
Versionierung und Freigabe	11	Content-Versionen und drei PENDING-Freigaben speichern
Status	12–13	Kanalstatus und Jobstatus aktualisieren
Abschluss	14	Zusammenfassungsdaten für Test und Monitoring zurückgeben

5.2 Konsistenz und Validierung

- Der OpenAI-Node (07) wird für jedes der drei Kanal-Items ausgeführt; alle Aufrufe greifen auf denselben Produktionsbrief zurück. Dadurch bleiben die Kanaltexte inhaltlich konsistent.
- Node 07 fordert für den Content-Body einen Zielbereich von 350 bis 1100 Zeichen. Node 07b sichert diesen Bereich technisch ab. Node 08 validiert anschließend pro Kanal Pflichtfelder, JSON-Struktur und eine harte Zeichenlänge von 300 bis 1200 Zeichen. Node 10 stellt sicher, dass drei Kanaltexte vorliegen und normalisiert die Kanal-IDs für Node 11.
- Versionen werden in `content.content_versions` persistiert und liegen damit nicht nur als flüchtige n8n-Ausgabe vor.
- Jeder Kanal erhält einen eigenen Freigabedatensatz; der Review-Status ist dadurch je Kanal nachvollziehbar.

6 | Human-in-the-Loop-Review

Der Human-in-the-Loop-Ansatz bindet menschliche Urteilskraft gezielt in KI-gestützte Abläufe ein [4]. In PA2 ist dieser Ansatz als Freigabeschritt nach der Generierung umgesetzt. Die menschliche Freigabe ist dabei kein optionaler Schritt der Benutzeroberfläche, sondern ein technisches Gate: Nach der Produktion stehen die drei Freigaben auf PENDING. Eine verantwortliche Person prüft die Inhalte in NocoDB und setzt den Freigabestatus je Kanal (Abbildung 6.1).

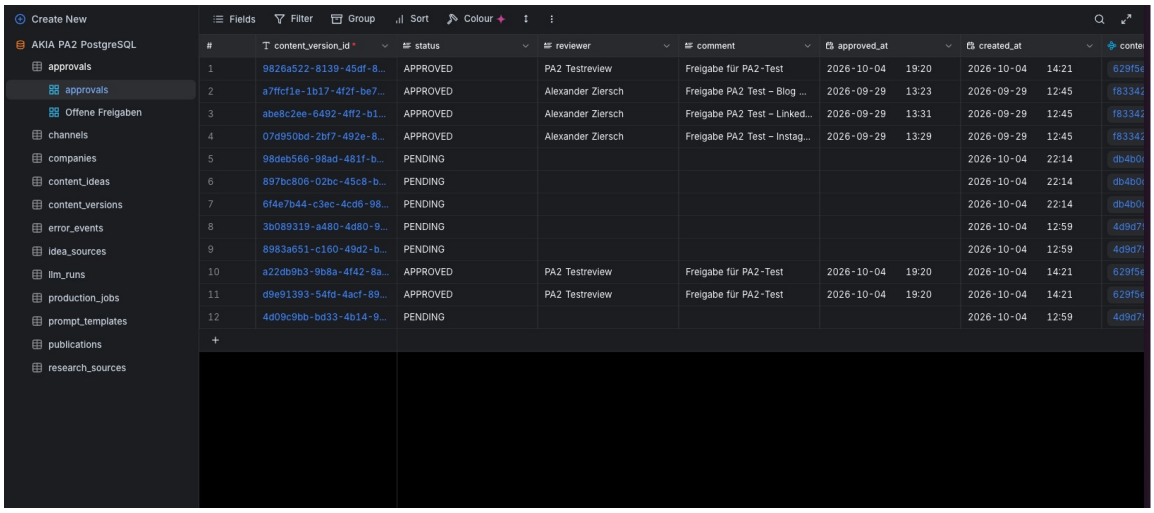


Abbildung 6.1: NocoDB-Ansicht mit Freigaben in den Status APPROVED und PENDING

Tabelle 6.1: Freigabezustände und ihre Folgen

Zustand	Bedeutung	Folge
PENDING	noch nicht geprüft	Das Gate in Node 17b bleibt geschlossen; Workflow C endet vor Node 18.
APPROVED	vom Reviewer freigegeben	Bei 3 × APPROVED wird die Publishing-Vorbereitung gestartet.
REJECTED	nicht freigegeben	Keine Publishing-Vorbereitung. Ein automatisierter Überarbeitungszyklus ist noch nicht implementiert (Kapitel 12).

Kontrollprinzip

Die KI liefert Entwürfe, trifft jedoch keine Veröffentlichungsentscheidung. Das System trennt Automatisierung und Verantwortung.

7 Workflow C – Review und Publishing

Der Workflow `PA2_Review_Publishing` besitzt einen manuellen Trigger für Tests und einen Schedule Trigger im 15-Minuten-Raster. Node 17a wählt dynamisch den nächsten Job aus, der den Status COMPLETED hat, dessen drei Freigaben auf APPROVED stehen und für dessen aktuelle Instagram-Version noch kein Publikationsdatensatz existiert. Bereits übernommene Jobs werden dadurch nicht erneut ausgewählt.



Abbildung 7.1: Finaler Review- und Publishing-Workflow mit Live-Veröffentlichung über Buffer

Tabelle 7.1: Nodes von Workflow C in Ausführungsreihenfolge

Node	Funktion
17a Freigabestatus laden	Dynamische Warteschlange gemäß dem oben genannten Auswahlkriterium
17b Alle freigegeben?	Boolesches Gate `all_approved`
18 Publish-Vorbereitung	Erzeugt ein einheitliches Array `publish_items` für Blog, Instagram und LinkedIn.
23 Bildgenerierung	GPT Image erzeugt ein quadratisches Instagram-Bild [5].
23b Cloudinary	Lädt das Bild hoch und liefert eine `secure_url` [6].
24 Instagram via Buffer	Sendet Text und öffentliche Cloudinary-Bild-URL an die Buffer GraphQL API; Test zunächst mit `addToQueue`, anschließend live mit `shareNow` [2].
20 Monitoring	`INSERT`/`UPDATE` in `content.publications`; Buffer-Post-ID als `external_id`; Queueing = SCHEDULED, Live-Post = PUBLISHED
19 Benachrichtigung	Gmail versendet eine Statusmail zur abgeschlossenen Publishing-Vorbereitung.

Begründung für Cloudinary und Buffer. Buffer erwartet für einen Bildpost eine öffentlich erreichbare Bild-URL im `assets`-Argument der GraphQL-Mutation `createPost` [2]. Das von GPT Image erzeugte PNG liegt zunächst binär in n8n vor und wird deshalb bei Cloudinary gespeichert. Die resultierende `secure_url` wird zusammen mit dem Instagram-Text an Buffer übergeben. Die Buffer-Authentifizierung erfolgt per Bearer-API-Key [3].

Nebenläufigkeit. Der Schedule Trigger startet alle 15 Minuten. Dauert ein Lauf, etwa wegen der Bildgenerierung, länger als dieses Intervall, könnte ein zweiter Lauf denselben Job auswählen, bevor der Publikationsdatensatz angelegt ist. Eine atomare Reservierung schließt dies aus, z. B. per `SELECT…FORUPDATESKIPLOCKED` [7] oder durch sofortiges Setzen eines Reservierungsstatus in Node 17a (Kapitel 12).

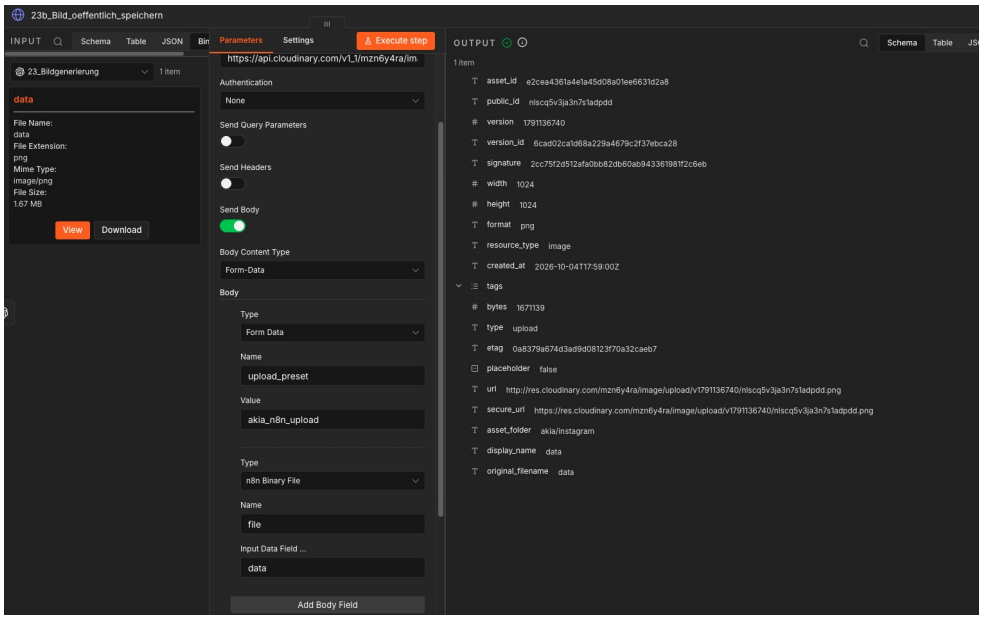


Abbildung 7.2: Der Cloudinary-Upload liefert eine öffentlich erreichbare URL

PA2 Content-System – Publishing Status Job-ID: 629f5e0c-edf4-4bcf-800a-d6a571fcace2 Kanal: instagram Status: PENDING Publication-ID: 694396f1-22a9-40ee-844d-2ca51ce00138 Instagram Post-ID: Noch nicht veröffentlicht Veröffentlicht am: Noch offen Versuche: 0 Letzter Fehler: Kein Fehler

---
This email was sent automatically with n8n
https://n8n.io

Abbildung 7.3: Empfangene n8n-Statusmail; der gezeigte Nachweis stammt aus einem früheren Publishing-Test

8 Zentrales Error-Handling

`PA2_Error_Handling` ist als eigenständiger Workflow veröffentlicht und in den Workflow-Einstellungen der Workflows A bis C als Error-Workflow hinterlegt. Ein solcher Workflow beginnt mit einem Error Trigger und wird ausgeführt, wenn eine Ausführung fehlschlägt [8]. Der Trigger erhält Metadaten der fehlgeschlagenen Ausführung, darunter Ausführungs-ID, zuletzt ausgeführter Node, Fehlermeldung und Workflow-Angaben [9].

Der Workflow normalisiert diese Daten und protokolliert sie in `content.error_events` (Node 15). Node 16 entscheidet anhand der Klassifizierung, ob ein Fehler als wiederholbar gilt: Wiederholbare Fehler laufen in den Zweig `16b_Wait_vor_Retry`, alle übrigen in den Final-Fail-Pfad `16c_Final_Fail_setzen` (Abbildung 8.1).

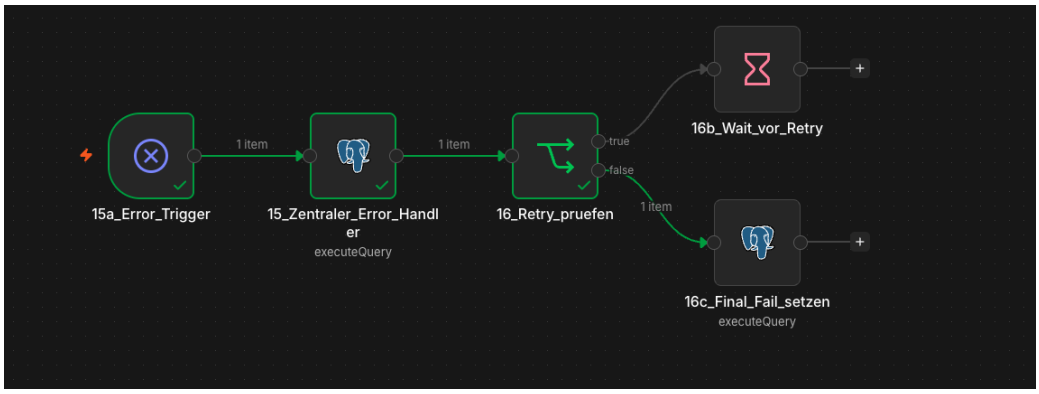


Abbildung 8.1: Zentraler Error-Workflow mit Retry-Prüfung und Final-Fail-Pfad

8.1 Umgesetzter Funktionsumfang

- Fehlerdaten werden mit `workflow_name`, `node_name`, `error_type` und `error_message` gespeichert.
- Fehler werden als wiederholbar klassifiziert; der Wiederholungszähler `retry_count` wird fortgeschrieben.
- Nicht wiederholbare bzw. endgültige Fehler werden im Final-Fail-Pfad behandelt.
- Ist im Fehlerkontext eine `job_id` vorhanden, wird der zugehörige Produktionsjob auf `FAILED` gesetzt und `error_count` erhöht. Fehler ohne eindeutig zuordenbaren Job werden unabhängig davon in `content.error_events` protokolliert.

Grenzen der Retry-Strategie

Der Error-Workflow läuft als eigene Ausführung und erhält lediglich Metadaten der fehlgeschlagenen Ausführung [9]; eine erneute Ausführung des fehlgeschlagenen Nodes ist darin nicht vorgesehen. Wiederholungsversuche auf Node-Ebene stellt n8n über die Node-Einstellung „Retry On Fail“ bereit [10]. Der Zweig `16b_Wait_vor_Retry` ist daher eine vorbereitete Strategie, aber kein vollständiger Replay-Mechanismus. Für externe API-Nodes sollte „Retry On Fail“ zusätzlich aktiviert werden.

9 Teststrategie und Nachweise

Die Tests wurden schrittweise durchgeführt: zunächst an einzelnen Nodes (N), dann an vollständigen Workflows (W) und abschließend an der Kette über alle Workflows (S). Dabei wurden reale Datenbankeinträge erzeugt und gegen den erwarteten Status geprüft. Tabelle 9.1 ordnet jedem Testfall eine Ebene und einen Nachweis zu.

9.1 Negativtests

Ein Gate gilt erst als nachgewiesen, wenn auch der Negativfall geprüft ist. Die folgenden Testfälle sind in der vorliegenden Dokumentation nicht belegt. Wurden sie durchgeführt, sind Ergebnis und Nachweis zu ergänzen; andernfalls sind sie als Testlücke auszuweisen.

9.2 Beispiel: automatische Ende-zu-Ende-Kette

Schedule → KI-Idee → PostgreSQL (QUEUED) → Production-Webhook → Produktion (Nodes 02–14) → drei Freigaben (PENDING) → Human Review (APPROVED) → Review-Schedule (15 min) → Publish-Payload → Bild → Cloudinary → Buffer → Instagram live (PUBLISHED) → Gmail

Tabelle 9.1: Durchgeführte Testfälle

ID	Testfall	Ebene	Erwartung	Ergebnis	Nachweis
T01	KI-Idee erzeugen	N	strukturiertes JSON, Speicherung als QUEUED	bestanden	Abb. 4.1
T02	Production-Webhook	W	`idea_id` wird im Produktionsworkflow empfangen	bestanden	Abb. 4.2
T03	Gezielte Ideenauswahl	N	Node 02 lädt exakt die übergebene `idea_id`	bestanden	Abb. 4.2
T04	3-Kanal-Produktion	W	Blog, Instagram und LinkedIn vollständig	bestanden	Abb. 5.1
T05	Freigabedatensätze	N	drei neue PENDING-Datensätze je Job	bestanden	Abb. 6.1
T06	Human Review	W	3 × APPROVED öffnet das Gate	bestanden	Abb. 6.1
T07	Dynamische Review-Warteschlange	W	bereits verarbeitete Jobs werden nicht erneut ausgewählt	bestanden	Node-17a-Test
T08	Bildgenerierung	N	PNG-Binärdaten aus GPT Image	bestanden	Abb. 7.1
T09	Cloudinary	N	`secure_url` wird erzeugt	bestanden	Abb. 7.2
T10	Publikations-Monitoring	N	Buffer-Post-ID wird gespeichert; Queueing ergibt SCHEDULED, Live-Publishing PUBLISHED	bestanden	Node-20-Test
T11	Gmail	N	Statusmail wird versendet und empfangen	bestanden	Abb. 7.3
T12	Fehlerprotokollierung	W	Datensatz in `error_events`, Final Fail	bestanden	Abb. 8.1
T13	Ende-zu-Ende-Kette	S	vollständiger Durchlauf von A bis C	bestanden	Abb. 4.1–7.3
T14	Instagram-Live-Publishing	W	Veröffentlichung eines neu generierten Bildposts über Buffer auf Instagram	bestanden	Live-Test am 05.10.2026
T19	Öffentliche Prüfer-End-to-End-Demo	S	Website-Klick startet Demo-Orchestrator, bestehende Produktion und Review/Publishing; ein neues Bild erscheint live auf Instagram; Response kehrt zur Website zurück	bestanden	Abb. 10.2

Tabelle 9.2: Negativtests

ID	Testfall	Erwartung	Ergebnis
T15	REJECTED blockiert	Ein abgelehnter Kanal verhindert die Publishing-Vorbereitung.	nicht durchgeführt
T16	Teilfreigabe blockiert	2 × APPROVED + 1 × PENDING öffnet das Gate nicht.	nicht durchgeführt
T17	Ungültige KI-Ausgabe	Node 08 erkennt fehlende Pflichtfelder; der Job wird nicht freigabefähig.	nicht durchgeführt
T18	Doppelter Webhook-Aufruf	Dieselbe `idea_id` erzeugt keinen zweiten Job.	nicht durchgeführt

10 Öffentliche Website und Deployment

Die Projektdemo ist als statische Website auf dem vorhandenen Hetzner-Server bereitgestellt. nginx liefert die Seiten über die bestehende TLS-Konfiguration von `akia-projekte.de` aus. Die Website greift nicht auf interne n8n- oder PostgreSQL-Zugänge zu.

Prüfer-Live-Demo. Die dedizierte Prüfer-Demo ist umgesetzt und am 05.10.2026 erfolgreich Ende-zu-Ende getestet. Ein Klick auf der öffentlichen Demo-Seite erzeugt eine neue `request_id` und ruft `PA2_Demo_Instagram` auf. Dieser Orchestrator legt eine Demo-Idee an, startet den bestehenden Produktionsworkflow, wartet per Polling auf drei erzeugte Versionen und Freigabedatensätze, setzt ausschließlich für diesen Demo-Job die drei Freigaben automatisch auf APPROVED und startet anschließend gezielt den bestehenden Review-/Publishing-Workflow. Dieser erzeugt ein neues KI-Bild, lädt es zu Cloudinary, veröffentlicht es über Buffer mit `shareNow` auf Instagram, schreibt PUBLISHED ins Monitoring und liefert das Ergebnis an den Demo-Workflow zurück. Die Website erhält abschließend die Webhook-Response. Derselbe `request_id`-Wert kann serverseitig nicht doppelt verarbeitet werden; neue bewusste Klicks erzeugen neue IDs und sind mehrfach testbar. API-Keys und Backend-Credentials bleiben ausschließlich serverseitig.



Abbildung 10.1: Klassische Projektansicht der AKIA-Demo

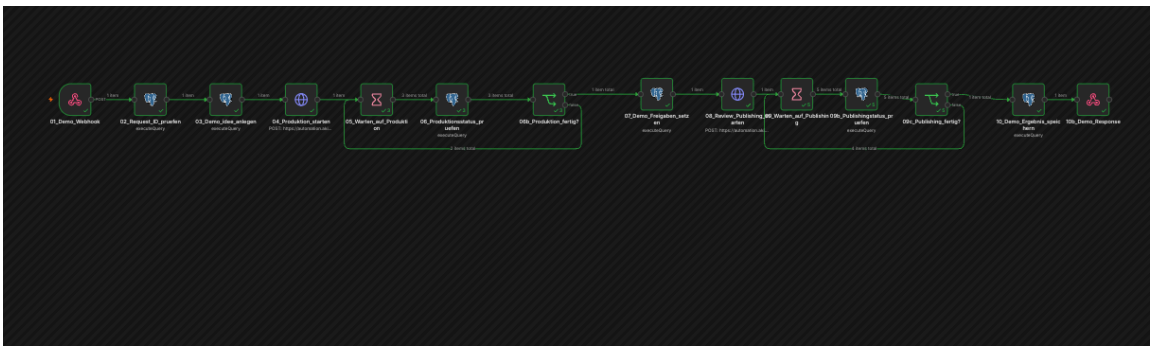


Abbildung 10.2: Erfolgreicher Prüfer-End-to-End-Lauf: Produktion und Publishing werden über den Demo-Orchestrator bis zur Response gekoppelt

Tabelle 10.1: Bereitgestellte Ansichten der Projektdemo

URL	Rolle
https://www.akia-projekte.de/projektaufgabe2/	Auswahlseite
https://www.akia-projekte.de/projektaufgabe2/klassisch/	Prüf- und Informationsansicht
https://www.akia-projekte.de/projektaufgabe2/3d/	Interaktive 3D-Perspektive
https://www.akia-projekte.de/projektaufgabe2/cyberflug/	Immersive Projektreise
https://www.akia-projekte.de/projektaufgabe2/demo-instagram/	Mehrfach testbare Prüfer-End-to-End-Live-Demo

10.1 Deployment-Nachweis

- Die Dateien liegen unter `/opt/akia-projekte/www/projektaufgabe2/`.
- Die nginx-Konfiguration wurde mit `nginx -t` erfolgreich geprüft.
- Alle fünf öffentlichen PA2-Einstiege lieferten serverseitig den Statuscode HTTP 200 (OK).
- Die bestehende Hauptseite und Projektaufgabe 1 wurden nicht überschrieben.

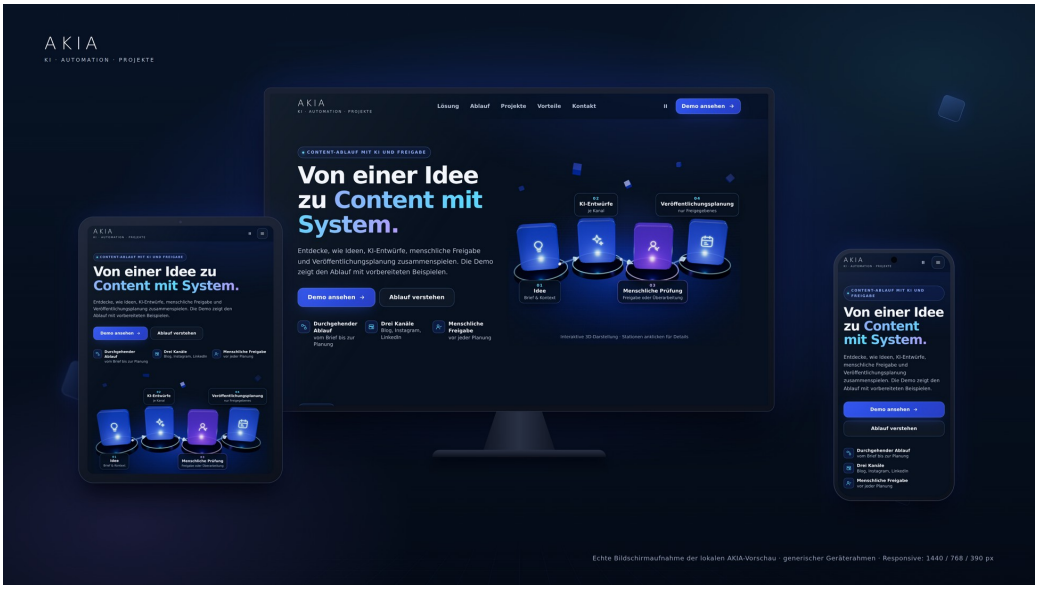


Abbildung 10.3: Responsive Darstellung der Projektseite

11 Sicherheits- und Betriebsaspekte

Tabelle 11.1: Sicherheits- und Betriebsaspekte

Thema	Umsetzung / Bewertung
Trennung Demo / Backend	Die öffentlichen Informationsansichten liefern keine Secrets aus. Nur die klar gekennzeichnete Prüfer-Live-Demo ruft einen dedizierten n8n-Webhook mit festem Testpfad auf.
HTTPS	nginx mit bestehenden Let's-Encrypt-Zertifikaten
Administrativer Zugriff	Administrativer SSH-Zugriff wurde über das private Tailscale-Netz durchgeführt; ein SSH-Zugriff über die öffentliche Server-IP war vom Testclient nicht erreichbar.
Secrets	Der Stack verwendet Docker- bzw. Datei-Secrets für Datenbank- und n8n-Komponenten.
n8n / NocoDB	Die Oberflächen sind über HTTPS-Domains erreichbar und durch die jeweilige Anwendungsauthentifizierung geschützt; Zugangsdaten werden nicht in der öffentlichen Projektwebsite ausgeliefert.
Human Review	Die Publishing-Vorbereitung startet nur bei vollständiger Freigabe.
Deduplizierung der Warteschlange	Node 17a überspringt Jobs, für die bereits ein Publikationsdatensatz existiert.
Production-Webhook	Derzeit ohne Authentifizierung. Risiko: Wer die URL kennt, kann kostenpflichtige KI-Läufe auslösen. Maßnahme: Header- oder JWT-Authentifizierung des Webhook-Nodes [11].
Datenübermittlung	Ideen- und Kontextdaten werden zur Generierung an OpenAI als externen Anbieter übermittelt. Für Echtdaten sind Datenschutz- und Auftragsverarbeitungsfragen zu klären.
Öffentliche Website	Impressum und Datenschutzerklärung müssen vor einem dauerhaften externen Betrieb mit den finalen Betreiberangaben vervollständigt werden.

12 Offene Punkte und Weiterentwicklung

Der Kern des Systems ist funktionsfähig und getestet. Die verbleibenden Punkte betreffen vor allem externe Plattformanbindungen und die Härtung für einen dauerhaften Produktivbetrieb.

Tabelle 12.1: Offene Punkte nach Priorität

Priorität	Punkt	Empfehlung
Mittel	Prüfer-Live-Demo weiter härten	Die `request_id`-Deduplizierung ist umgesetzt; für den Dauerbetrieb zusätzlich serverseitiges Rate-Limiting und optional eine signierte Demo-Anforderung ergänzen.
Hoch	Webhook-Härtung	Production-Webhook mit Header- oder JWT-Authentifizierung absichern [11].
Hoch	Negativtests	Testfälle T15–T18 durchführen und dokumentieren.
Mittel	Retry-Strategie	„Retry On Fail“ bei OpenAI-, HTTP-, Cloudinary- und Buffer-Nodes aktivieren [10]; den zentralen Handler für Protokollierung und Final Fail beibehalten.
Mittel	Nebenläufigkeit Workflow C	Job in Node 17a atomar reservieren, z. B. per `FORUPDATESKIPLOCKED` [7].
Mittel	Parametrisierte Abfragen	SQL-Abfragen in den Postgres-Nodes mit Query-Parametern statt Ausdrucksverkettung aufbauen; n8n weist in Node 21b selbst darauf hin (Abb. 4.1).
Mittel	Ablehnungspfad	Bei REJECTED eine gezielte Überarbeitungsversion und einen erneuten Freigabezyklus ergänzen.
Mittel	Rechtliche Angaben der Website	Finale Impressums- und Datenschutzangaben hinterlegen.
Niedrig	Reporting	Dashboard für Durchlaufzeiten, Fehlerquote, Freigabezeiten und Kanaloutput ergänzen.

13 Fazit

PA2 ist als modulares, nachvollziehbares Automationssystem aus vier fachlichen Kernworkflows plus einem separaten Prüfer-Demo-Orchestrator umgesetzt. Die wesentliche Stärke liegt weniger in der KI-Textgenerierung als in der Orchestrierung: strukturierte Datenhaltung, gezielte Webhook-Übergabe, technische Validierung, Versionierung, Human-in-the-Loop und ein separater Prozess für die Publishing-Vorbereitung.

Der Kernprozess ist vollständig implementiert und durch Tests belegt. Fünf n8n-Workflows sind veröffentlicht: vier Kernworkflows und der Demo-Orchestrator. Die öffentliche Website stellt das Projekt in drei Darstellungsvarianten bereit und hebt zusätzlich eine Prüferansicht sowie die echte Live-Demo hervor. Die direkte Veröffentlichung auf Instagram ist über Buffer aktiv. Zusätzlich wurde der öffentliche Website-Klick erfolgreich bis zu einem neuen Instagram-Bildpost und zurück zur Webhook-Response Ende-zu-Ende getestet. Tabelle 13.1 bilanziert die Zielerreichung bezogen auf die Projektziele aus Kapitel 1.

Tabelle 13.1: Zielerreichung

Ziel	Kurzbeschreibung	Erreicht	Nachweis
Z1	Idee als strukturierter Datensatz	ja	T01
Z2	Getrennte Kanaltexte aus gemeinsamem Brief	ja	T04
Z3	Validierung, Versionierung, Freigabeobjekt je Kanal	ja	T04, T05
Z4	Kein Publishing ohne Freigabe	ja (Positivfall)	T06; Negativfälle T15/T16 offen
Z5	Automatische Vorbereitung, Bild, Live-Veröffentlichung und Protokoll	ja	T08–T11, T14
Z6	Zentrale Fehlererfassung, Jobstatus	ja; Retry-Härtung offen	T12
Z7	Öffentliche Demo ohne Backend-Zugriff	ja	Kap. 10

Gesamtbewertung

Das System erfüllt die Kernziele eines automatisierten Content-Workflows mit menschlicher Kontrolle und nachvollziehbarer Speicherung. Die tatsächliche Instagram-Veröffentlichung (Z5) sowie der öffentlich ausgelöste Prüfer-End-to-End-Pfad sind umgesetzt und live getestet. Für einen dauerhaften Realbetrieb bleiben vor allem Rate-Limiting bzw. zusätzliche Webhook-Härtung, die noch offenen Negativtests und eine zusätzliche Retry-Härtung sinnvoll.

Quellenverzeichnis

[1] „PostgreSQL 17 Documentation," The PostgreSQL Global Development Group, besucht am 5. Okt. 2026. Adresse: https://www.postgresql.org/docs/17/.

[2] „Create Image Post," Buffer API Documentation, Buffer, Inc., besucht am 5. Okt. 2026. Adresse: https://developers.buffer.com/examples/create-image-post.html.

[3] „Authentication," Buffer API Documentation, Buffer, Inc., besucht am 5. Okt. 2026. Adresse: https://developers.buffer.com/guides/authentication.html.

[4] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán und Á. Fernández-Leal, „Human-in-the-loop machine learning: a state of the art," *Artificial Intelligence Review*, Jg. 56, Nr. 4, S. 3005–3054, 2023. DOI: 10.1007/s10462-022-10246-w.

[5] „Image generation," OpenAI API Documentation, OpenAI, besucht am 5. Okt. 2026. Adresse: https://developers.openai.com/api/docs/guides/image-generation.

[6] „Upload API Reference," Cloudinary Documentation, Cloudinary Ltd., besucht am 5. Okt. 2026. Adresse: https://cloudinary.com/documentation/image_upload_api_reference.

[7] „SELECT – The Locking Clause," PostgreSQL 17 Documentation, The PostgreSQL Global Development Group, besucht am 5. Okt. 2026. Adresse: https://www.postgresql.org/docs/17/sql-select.html.

[8] „Handle errors gracefully," n8n Docs, n8n GmbH, besucht am 5. Okt. 2026. Adresse: https://docs.n8n.io/build/flow-logic/handle-errors-gracefully/.

[9] „Error Trigger," n8n Docs, n8n GmbH, besucht am 5. Okt. 2026. Adresse: https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.errortrigger/.

[10] „Work with nodes," n8n Docs, n8n GmbH, besucht am 5. Okt. 2026. Adresse: https://docs.n8n.io/build/understand-workflows/workflow-components/work-with-nodes/.

[11] „Webhook," n8n Docs, n8n GmbH, besucht am 5. Okt. 2026. Adresse: https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.webhook/.

Anhang A | Kernstatus und Prüfcheckliste

Tabelle A.1: Prüfcheckliste zum Systemstand

Prüfpunkt	Status
`PA2_Automations_Ideen` veröffentlicht	ja
`PA2-Produktion` veröffentlicht	ja
`PA2_Review_Publishing` veröffentlicht	ja
`PA2_Error_Handling` veröffentlicht	ja
`PA2_Demo_Instagram` veröffentlicht	ja
Production-Webhook erreichbar	ja
drei Kanaltexte je Job	ja
drei Freigaben je Job	ja
Human Review getestet	ja (Positivfall)
Bildgenerierung getestet	ja
Cloudinary getestet	ja
Publikations-Monitoring getestet	ja
Gmail getestet	ja
Website online	ja
Instagram-Live-Publishing über Buffer	ja, live getestet
Öffentliche Prüfer-E2E-Demo	ja, Website-Klick bis neuem Instagram-Bild und Response erfolgreich

Hinweis zur Dokumentation

Die Abbildungen zeigen den getesteten Systemstand vom 04. und 05. 10. 2026. UUIDs, IDs und Zeitstempel sind technische Testdaten und dienen ausschließlich als Nachweis der tatsächlichen Verarbeitung. Zugangsdaten, API-Secrets und private Backend-Credentials sind nicht Bestandteil dieser Dokumentation.

Anhang B | Modelle und Prompt-Vorlagen

Für die Reproduzierbarkeit werden die im Projekt verwendeten Modelle, Parameter und Prompt-Vorlagen dokumentiert. Änderungen an verfügbaren Modellnamen oder API-Versionen können zukünftig Anpassungen erfordern.

Tabelle B.1: Eingesetzte Modelle und Parameter

Verwendung	Modell / Parameter
Ideengenerierung (Node 21)	GPT-6-LUNA; textbasierte strukturierte JSON-Ausgabe gemäß Prompt
Kanaltexte (Node 07)	GPT-6-LUNA; JSON mit `title`, `body`, `hashtags`, `cta`; Body-Zielbereich 350--1100 Zeichen
Bildgenerierung (Node 23)	GPT-Image; 1024×1024 Pixel; Qualitätsstufe High

Prompt-Prinzipien. Node 21 verwendet einen System-Prompt für einen deutschsprachigen Content-Strategen und fordert genau eine neue Content-Idee als JSON mit Titel, Idee, Keywords, Content-Ziel, Priorität und Herkunft. Node 07 verwendet einen System-Prompt für einen professionellen deutschsprachigen Content-Redakteur, berücksichtigt Unternehmen, Branche, Kanal und Content-Ziel und fordert ausschließlich valides JSON in der Form `title`, `body`, `hashtags`, `cta`. Der Body wird bereits im Prompt auf 350–1100 Zeichen begrenzt und anschließend technisch durch Node 07b/08 abgesichert. Die Struktur wird im Projekt über Promptvorgabe und nachgelagerte JavaScript-Validierung erzwungen; Structured Outputs sind nicht Voraussetzung.