NEG AG

NORTHGATE ENTERPRISES GROUP
AKTIENGESELLSCHAFT

EXPOSE AUTOMATON

Technical Architecture & Developer Documentation

Version 2.0 | March 2026 | CONFIDENTIAL

Prepared for: Development Team

1. System Overview

The Expose Automaton is an end-to-end pipeline that transforms raw property files (photos, documents, floor plans) from a OneDrive folder into a professionally branded, print-ready PDF expose for NEG AG. The system handles two output formats: Short Form (portrait A4, compact) and Long Form (landscape A3 spread, booklet-style).

1.1 Architecture Diagram

The system follows this linear pipeline:

```
OneDrive Folder  -->  [W-01: Ingestion]  -->  NocoDB (3 tables)
                                                      |
                                                  UI Form (React)
                                                  User edits/approves
                                                      |
                                              [W-02: PDF Generation]
                                                      |
                                              Gotenberg (HTML->PDF)
                                                      |
                                                  Final PDF output
```

1.2 Tech Stack

Layer	Technology	Purpose
Orchestration	n8n (self-hosted)	Workflow automation for both W-01 and W-02
Database	NocoDB	3 tables: Exposes, Sections, Images
AI Vision	Google Gemini	Image classification, quality scoring, metadata extraction
OCR/Text	Azure Document Intelligence	PDF/DOCX/XLSX text extraction
Content AI	LLM (via n8n)	Template generation, section content writing
Frontend	React + Vite	Review/edit UI at expose.neg.ag
API Proxy	Express (server.cjs)	NocoDB token injection, webhook forwarding
PDF Engine	Gotenberg	Headless Chrome HTML-to-PDF conversion
File Storage	Microsoft OneDrive	Source files + generated PDFs
Deployment	Docker	Containerized UI + proxy

2. Database Schema (NocoDB)

Three tables form the data backbone. All are linked by ExposeId. The NocoDB base ID is pk7i4bqetoy32x4.

2.1 Exposes Table

```
Table ID: mv00nvrs79hcarh
```

Field	Type	Description
Id	Auto ID	Primary key
Title	String	Property title (from AI or manual)
PropertyId	String	Internal property reference, e.g. LEI-2025-001
Address, City, ZipCode	String	Property location fields
Status	SingleSelect	Lifecycle: Draft > AI_Draft > Manual_Review > Submitted > generating_pdf > completed
ExposeType	**NEW FIELD**	short_form | long_form (determines PDF template)
AIConfidenceScore	Number	0-100, set by AI during ingestion
ProcessingStatus	String	AI_Draft or Manual_Review (drives UI banner)
OneDriveSourcePath	String	Original OneDrive folder path
OneDriveFolderId	String	OneDrive folder ID for re-access
ExtractedFieldsJson	LongText	JSON array of {label, value} from AI extraction
Latitude, Longitude	Number	Geocoded coordinates
expose_pdf	Attachment	Generated PDF file attachment

2.2 Sections Table

```
Table ID: m0ckfqsmhat77di
```

Field	Type	Description
ExposeId	String (FK)	Links to Exposes.Id
SectionType	String	cover, key_facts, description, text_content, location_macro, location_micro, city_stats, market_analysis, image_gallery, floor_plans, data_table, contact, foreword, legal_disclaimer, references, press_quotes, impressum
Heading	String	Section display title
Content	LongText	Plain text or JSON depending on section type
Order	Number	Display/render order
ImageLayout	String	Hero + 2 Subs, Grid, etc.
ImageRefsJson	LongText	JSON array of NocoDB Image IDs linked to this section
Enabled	**NEW FIELD**	Boolean (default: true). If false, section is skipped in PDF generation but kept in DB for editing.
AIConfidence	Number	Per-section AI confidence score

2.3 Images Table

```
Table ID: mto0jc7j6zzf6nr
```

Field	Type	Description
ExposeId	String (FK)	Links to Exposes.Id
SectionId	String (FK)	Optional: links to Sections.Id
FileName	Attachment	NocoDB attachment (array with signedPath, thumbnails)
Role	String	Hero or Sub
AICategory	String	Exterior, Interior, Kitchen, Floor Plan, Aerial, Site Map, etc.
QualityScore	Number	AI-assigned 0-100 quality rating
ContentType	String	photograph, rendering, floor_plan, map, document_scan
FloorPlanLevel	String	EG, 1.OG, 2.OG, KG, DG, etc.
OneDriveFileId	String	Original OneDrive file ID for re-download

3. Workflow W-01: Data Ingestion

Triggered: Manually from UI (user selects OneDrive folder) or on schedule. This workflow scans a OneDrive folder, classifies all files with AI, extracts property data, and creates structured records in NocoDB.

3.1 Pipeline Steps

1. List Inbox Folders: Calls Microsoft Graph API to list folders under NEG (company doc)/Projekte/4_Expose/
2. Filter New Folders: Removes folders prefixed _done_ or _processing_, cross-checks NocoDB for duplicates via OneDriveFolderId.
3. List All Files: Enumerates every file in the selected folder.
4. Categorize Files: Sorts into images (jpg/png/webp), documents (pdf/docx/xlsx), floor plans (keyword detection), and unprocessable (CAD/archives).
5. AI Image Analysis: Each image sent to Gemini Vision. Returns: category, content_type, quality_score, is_hero_candidate, detected_text, german description, temporal status.
6. Image Aggregation: Combines results, selects hero image (highest quality hero candidate), computes section flags (has_exterior, has_floor_plans, etc.).
7. Document Text Extraction: PDFs/DOCX via Azure Document Intelligence, simple text files read directly.
8. Geocoding: Best address geocoded via Google. Fetches Mikrolage POIs (5km) and Makrolage POIs (50km), map URLs, Street View.
9. AI Template Generation: All data fed to LLM. Outputs structured JSON with property metadata, key facts, and ordered section array. All values German, keys English.
10. Template Validation: Mandatory sections checked (cover, key_facts, contact). Text stored as strings, locations get real POI data, data tables validated.
11. Database Persistence: Creates Expose record, Section records (one per section), Image records (one per image) in NocoDB.
12. Cross-Linking: Patches ImageRefsJson on sections with real NocoDB image IDs. Links map images to location sections.
13. Email Notification: Sends branded email to va@saits.ai with AI confidence %, counts, and link to UI form.

3.2 What Needs to Change

- **Manual Trigger:** W-01 currently auto-polls. Needs a new webhook entry point that accepts { folder_id, folder_name } from the UI. The auto-poll can remain as fallback.
- **ExposeType Field:** The ingestion should default ExposeType to 'long_form' (or accept it from the UI trigger payload) and save it on the Expose record.
- **Enabled Default:** All sections created by W-01 should have Enabled: true by default.

4. Workflow W-02: PDF Generation

Triggered: POST webhook to /expose-approved with { expose_id }. Fetches all data, assembles HTML, converts to PDF via Gotenberg.

4.1 Current Pipeline (Short Form)

14. Webhook: Receives expose_id.
15. Update Status: Sets status to generating_pdf in NocoDB.
16. Fetch Data: Gets Expose record, all Sections (sorted by Order), all Images.
17. Download Images: Loops through images, fetches actual files from NocoDB attachment URLs.
18. Merge: Combines expose data + sections + image binaries.
19. Map Images to Sections: Three strategies: ImageRefsJson IDs > SectionId field > AI category/role fallback.
20. Assemble HTML: Builds complete self-contained HTML using Short Form template (portrait A4, wave curves, vertical layout).
21. Convert to PDF: Sends HTML to Gotenberg (headless Chrome). Output: portrait A4 PDF.
22. Update Status: Sets status to completed, attaches PDF to expose record.
23. Respond to Webhook: Returns success.

4.2 What Needs to Change for Long Form

This is the biggest piece of new work.

- **Branching:** After step 3 (Fetch Data), add an IF node: if ExposeType === 'long_form' go to Long Form assembler, else keep existing Short Form assembler.
- **New HTML Assembler:** A new Code node that builds the Long Form landscape HTML. Must use the CSS from test-template.html with page size 420mm x 297mm.
- **Section Filtering:** The assembler must skip sections where Enabled === false.
- **Gotenberg Config:** Change page size to 420mm x 297mm for Long Form. Short Form stays at A4 portrait.
- **Dynamic TOC:** The Long Form generates a Table of Contents page automatically from enabled sections with computed page numbers.

4.3 Long Form Page Types & CSS Classes

Page	CSS Class	Layout	Status
Cover	.cover	Hero + brand panel	DONE
Table of Contents	.idx	50/50 spread	DONE
Foreword	.fw	60/40 text+photo	DONE
Legal Disclaimer	.legal	Full-width multi-col	TO BUILD
Key Facts	.kf	50/50 table+image	DONE
Planning Variants	.proj	50/50 text+image	TO BUILD
Location Macro	.loc	35/65 text+map	DONE
POI Photos	.photos	3-col full-bleed	TO BUILD
Location Micro	.loc-micro	Photo+map+legend	DONE
Full-bleed Image	.spread-full	Full spread photo	TO BUILD

City Stats	.city	60/40 text+photos	**DONE**
Floor Plans	.fp	Table + plan image	**PARTIAL**
Floor Plan Detail	.fp-detail	2-up with rooms	**TO BUILD**
Initiator	.text-split	50/50 text+image	**DONE**
References	.refs	Photos + bullet list	**TO BUILD**
Press Quotes	.press	Quotes layout	**TO BUILD**
Impressum	.impressum	Legal footer	**TO BUILD**
Contact	.contact	55/45 Z-pattern	**DONE**

5. UI Form (React Application)

The UI is a React + Vite app deployed via Docker. It has two main views: List View (all exposes) and Form View (edit/review a single expose). It communicates with NocoDB through an Express proxy that injects the API token server-side.

5.1 Current Component Tree

```
App.jsx
  +-- ExposeList.jsx          (list view, status pills, AI scores)
  +-- AISuggestionBanner.jsx   (Accept All / Review banner)
  +-- SectionText.jsx          (text content sections)
  +-- SectionImages.jsx         (image gallery with Hero/Sub)
  +-- SectionKeyValue.jsx       (key facts, contact info)
  +-- SectionLocation.jsx       (maps + POI tables)
  +-- SectionCityStats.jsx      (city data + stats)
  +-- SectionDataTable.jsx      (tabular data)
  +-- SectionFeatures.jsx       (feature lists)
  +-- api/nocodb.js             (API layer + proxy config)
```

5.2 What Needs to Change

5.2.1 OneDrive Folder Browser (NEW)

- **New API endpoint:** GET /api/onedrive/folders in server.cjs. Calls Microsoft Graph API to list folders under the expose path. Returns { id, name, createdDate, fileCount }.
- **New component:** OneDriveFolderPicker.jsx. Shows a list/grid of available OneDrive folders. User selects one and clicks Generate.
- **Trigger flow:** On click, POST to /api/webhook/ingest-folder with { folder_id, folder_name }. This triggers W-01 for that specific folder.
- **OAuth:** The Express proxy needs a Microsoft Graph access token. Options: share the n8n OAuth credential (extract refresh token), or register a separate Azure AD app.

5.2.2 Expose Type Selector (NEW)

- Add a dropdown/toggle at the top of the form: Kurz-Expose (Short Form) | Lang-Expose (Long Form).
- Stored in Exposes.ExposeType field. Default: long_form for new ingestions.
- When type changes, show/hide Long Form specific section types (Foreword, Legal, References, etc.).
- Pass expose_type in the webhook payload when triggering W-02.

5.2.3 Section Enable/Disable (NEW)

- Each section card gets a toggle/checkbox: Include in PDF.
- Maps to Sections.Enabled field (boolean, default true).
- Disabled sections stay in the form but are grayed out. They are excluded from the PDF generation HTML assembly.
- The submit/update API call must save the Enabled value for each section.

5.2.4 New Section Types for Long Form

- Add buttons: + Vorwort, + Rechtlicher Hinweis, + Referenzen, + Pressestimmen, + Impressum.
- These only appear when ExposeType === 'long_form'.
- Each maps to a new SectionType in the DB and a corresponding React component.

6. Long Form HTML Template

The template is at Long Form/html/test-template.html. It uses landscape A3 spreads (420mm x 297mm) with the NEG AG brand system.

6.1 Brand Specifications

Element	Value	CSS Variable
Primary Blue	#2B3A44	--blau
Gold Accent	#B8A387	--gold
Text Grey	#414D4D	--schrift
Heading Font	DM Serif Display (italic)	--heading
Body Font	Open Sans (300 weight)	--body
Page Size	420mm x 297mm (A3 landscape)	--page-w / --page-h
Z-Pattern	Diagonal stripes at -55deg, 6% white	.z-pattern-bg

6.2 Spread Layout System

Every page is a spread (two-page view). The base structure is:

```
<div class="page">                          <!-- 420x297mm -->
  <div class="spread">                      <!-- flexbox, h:100% -->
    <div class="spread-left">  ... </div>  <!-- 50% width -->
    <div class="spread-right"> ... </div>  <!-- 50% width -->
  </div>
</div>
```

Most sections use custom splits (35/65, 60/40, etc.) rather than the generic 50/50.

7. API Reference

7.1 Express Proxy (server.cjs)

All frontend API calls go through the Express proxy which injects the NocoDB API token server-side. The browser never sees the token.

Method	Path	Description
GET	/api/nocodb/*	Proxies to NocoDB API with token injection
POST	/api/nocodb/*	Proxies create/update requests to NocoDB
PATCH	/api/nocodb/*	Proxies patch requests to NocoDB
DELETE	/api/nocodb/*	Proxies delete requests to NocoDB
POST	/api/webhook/expose-approved	Forwards to n8n W-02 webhook
GET	**/api/onedrive/folders**	NEW: Lists OneDrive folders via Microsoft Graph
POST	**/api/webhook/ingest-folder**	NEW: Triggers W-01 for a specific folder

7.2 NocoDB Table IDs

```
Exposes:  mv00nvrs79hcarh
Sections: m0ckfqsmhat77di
Images:   mto0jc7j6zzf6nr
Base ID:  pk7i4bqetoy32x4
```

7.3 Webhook Payloads

W-02 Trigger (PDF Generation)

```
POST /api/webhook/expose-approved
{ "expose_id": 42 }
  --> NEW: also needs { "expose_type": "long_form" }
```

W-01 Manual Trigger (NEW)

```
POST /api/webhook/ingest-folder
{ "folder_id": "abc123", "folder_name": "Projekt XY", "expose_type": "long_form" }
```

8. Expose Status Lifecycle

```
[Draft]                    Manual creation in UI
   |
[AI_Draft]                 After W-01 completes (high confidence)
   |   OR
[Manual_Review]            After W-01 completes (low confidence)
   |
[Submitted]                User clicks 'Generate PDF' in UI
   |
[generating_pdf]           W-02 starts processing
   |
[completed]                PDF attached, ready for download
```

The UI shows different behaviors based on status:

- **AI_Draft / Manual_Review:** Shows AI Suggestion Banner with Accept All and Review Individually options.
- **Submitted:** Shows 'Eingereicht' pill. PDF generation in progress.
- **completed:** Shows 'Abgeschlossen' pill and PDF download button in header.

9. Project File Structure

```
Expose/
+-- Brand Identity/
|   +-- documents/                  Corporate identity + style guide PDFs
|   +-- extracted logos/             PNG logos (dark, white, line, icon)
+-- DB Schema/
|   +-- exposes_table.csv            Column definitions
|   +-- sections_table.csv           Column definitions
|   +-- images_table.csv             Column definitions
+-- Long Form/
|   +-- html/test-template.html      Current WIP landscape template
|   +-- Examples/NEG_AG_Layoutvorlage_200504.pdf   Target reference
+-- Short Form/
|   +-- html/expose-template-v3.html  Current production template
|   +-- Examples/                     Sample output PDFs
+-- N8N/
|   +-- W-01 Expose Data & Picture Ingestion.txt   n8n JSON export
|   +-- W-02 PDF Generation.txt                    n8n JSON export
+-- UI Form/
|   +-- src/App.jsx                  Main application
|   +-- src/components/              React section components
|   +-- src/api/nocodb.js            API layer
|   +-- server.cjs                   Express proxy
|   +-- Dockerfile                   Production build
+-- Fields.xlsx                      Field mapping reference
+-- Updated Layout.png               Handwritten layout sketch
```

10. Recommended Build Order

Follow this sequence. Each step depends on the previous:

#	Task	Files Affected	Effort	Depends On
1	Add ExposeType + Enabled fields to NocoDB	NocoDB admin	30 min	-
2	Complete Long Form HTML template (missing pages)	test-template.html	2-3 days	-
3	UI: Expose type selector + section toggles	App.jsx, components/*	1-2 days	1
4	UI: New Long Form section components	New components	1 day	3
5	W-02: Long Form HTML assembler + branching	W-02 n8n workflow	2-3 days	1, 2
6	W-02: Gotenberg landscape config	W-02 n8n workflow	1 hour	5
7	UI: OneDrive folder browser	server.cjs, new component	2 days	3
8	W-01: Manual trigger webhook endpoint	W-01 n8n workflow	2 hours	7
9	Integration testing: full flow end-to-end	All	2 days	All

Total estimated effort: 10-14 working days.