



ALL IN ONE REAL ESTATE SOFTWARE

AIORES PRODUKTDOKUMENTATION

Entwicklungsleitfaden

Aktueller Stand, Architektur und der Weg zur eigenständigen Servertechnik

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Überblick und Zielsetzung	3
2. Aktueller Entwicklungsstand.....	3
2.1 Bestehende Funktionsmodule	3
2.2 Begleitende Werkzeuge außerhalb der WebApp	3
3. Inhaltliche Struktur der Bausteine.....	5
3.1 Gemeinsame Stammdatenbasis	5
3.2 Einheitliches Zugriffs- und Authentifizierungskonzept	5
3.3 Marken- und Layoutkonzept der Kommunikation	5
3.4 Dokumentengenerierung als durchgängiges Prinzip	5
4. Technische Weiterverarbeitung der Inhalte	6
4.1 Automatisierte Dokumentenverarbeitung (Rechnungswesen).....	6
4.2 Geplante KI-gestützte Vorgangsbearbeitung	6
5. Strategischer Weg zur eigenständigen Servertechnik	7
5.1 Auslösekriterien für die Migration.....	7
5.2 Vorbereitende Maßnahmen im aktuellen Entwicklungsstadium	7
5.3 Zielarchitektur (konzeptionell)	7
6. Zusammenfassung.....	9

1. Überblick und Zielsetzung

AIORES (All In One Real Estate Software) ist eine im Aufbau befindliche Softwarelösung zur Optimierung administrativer Prozesse in der WEG- und Hausverwaltung. Das Projekt entsteht modular: Statt eine monolithische Anwendung von Beginn an vollständig neu zu entwickeln, werden einzelne, in der Praxis erprobte Funktionsbausteine schrittweise zusammengeführt und vereinheitlicht.

Dieser Leitfaden dokumentiert den aktuellen Entwicklungsstand, die inhaltliche Struktur der bestehenden Bausteine sowie deren technische Weiterverarbeitung. Darüber hinaus beschreibt er die strategische Zielsetzung: Sobald die einzelnen Bausteine ein Volumen und einen Reifegrad erreicht haben, der eine eigenständige Infrastruktur rechtfertigt, soll die Software von der aktuellen Basis auf eigene Servertechnik überführt werden.

2. Aktueller Entwicklungsstand

Die Entwicklung erfolgt aktuell auf Basis einer Google-Apps-Script-WebApp-Architektur. Dieser Ansatz wurde bewusst gewählt, um frühzeitig praxistaugliche Module mit geringem Infrastrukturaufwand bereitzustellen und Funktionsumfang sowie Nutzerakzeptanz iterativ zu validieren, bevor in eine eigenständige technische Infrastruktur investiert wird.

2.1 Bestehende Funktionsmodule

Die WebApp-Familie umfasst aktuell folgende, größtenteils eigenständig nutzbare Module, die über ein gemeinsames Stammdaten- und Zugriffssystem miteinander verbunden sind:

- Erinnerungsmanagement – automatisierter Versand von Erinnerungs-E-Mails mit CC-Logik und einheitlichem Markenauftritt
- Beschlussvorlagen – kategorisierte, wiederverwendbare Vorlagen für Eigentümerbeschlüsse
- Schlüsselmanagement – Erfassung von Ausgabe und Rückgabe objektbezogener Schlüssel
- Objektbegehungsmanagement – Dokumentation turnusmäßiger Objektbegehungen
- Objektübernahmen / Objektübergaben – strukturierte Erfassung bei Verwaltungswechsel
- Energieausweise – Verwaltung und Zuordnung von Energieausweisen je Objekt
- Zählerstände Nutzerwechsel – Erfassung von Zählerständen bei Mieter- oder Eigentümerwechsel
- Wohnungsabnahmeprotokoll – digitale Protokollierung von Wohnungsabnahmen mit PDF-Ausgabe
- Auftragsmanagement – Schaden- und Auftragsaufnahme mit Rückmeldemanagement und automatischer Fristenüberwachung

Insgesamt basiert das Ökosystem auf mindestens 19 Funktionsmodulen, die auf einer gemeinsamen Datenbasis aus zentralen Tabellenblättern aufsetzen, darunter Objekte, Dienstleister, Standardaufträge, Journal, Mitarbeiter, Zugriff und Konfiguration.

2.2 Begleitende Werkzeuge außerhalb der WebApp

Parallel zur WebApp-Familie existieren eigenständige Python-basierte Werkzeuge, die punktuelle, datenintensive Aufgaben übernehmen und perspektivisch in die zentrale Plattform integriert werden sollen:

- Rechnungsumbenennung und -erkennung – automatisierte Erkennung, Klassifizierung und Umbenennung eingehender Rechnungen mittels OCR, vollständig lokal und ohne Internetverbindung
- Filterung unklarer Rechnungen – automatische Aussortierung nicht eindeutig zuordenbarer Belege in einen separaten Nachbearbeitungsordner
- Kautionsabrechnung – GUI-gestützte Erstellung von Kautionsabrechnungen mit Word-/PDF-Vorlagen
- Zeiterfassung – lokales Tool zur Arbeitszeiterfassung mit Excel-Anbindung

3. Inhaltliche Struktur der Bausteine

Die einzelnen Module folgen einem einheitlichen inhaltlichen Aufbauprinzip, das eine spätere Zusammenführung erleichtert:

3.1 Gemeinsame Stammdatenbasis

Alle Module greifen auf dieselben zentralen Stammdaten zu, statt eigene, redundante Datensätze zu pflegen. Dies betrifft insbesondere Objekt-, Eigentümer-, Mieter- und Dienstleisterdaten. Diese Vereinheitlichung ist die wichtigste inhaltliche Voraussetzung für eine spätere Migration auf eine eigenständige Datenbankarchitektur, da sie Dateninkonsistenzen bereits im aktuellen Entwicklungsstadium vermeidet.

3.2 Einheitliches Zugriffs- und Authentifizierungskonzept

Ein durchgängiges Login- und Token-System regelt den Zugriff auf alle Module. PDF-Dokumente werden für angemeldete Nutzer direkt im Browser bereitgestellt, ohne zusätzliche Freigabeabfragen externer Speicherdienste – der gültige Zugriffslink gilt als ausreichender Berechtigungsnachweis. Dieses Konzept ist bewusst plattformunabhängig gestaltet, sodass es sich auf eine zukünftige Serverarchitektur übertragen lässt, ohne dass Nutzer eine Verhaltensänderung wahrnehmen.

3.3 Marken- und Layoutkonzept der Kommunikation

Auf das Erscheinungsbild der nach außen sichtbaren Kommunikation wurde von Beginn an besonderer Wert gelegt. E-Mails, die aus den Modulen heraus versendet werden, folgen einem einheitlichen, sorgfältig gestalteten Layout: konsistente Logoplatzierung (Logobreite 570 px für ein scharfes, aber kompaktes Erscheinungsbild), klare Typografie, ruhige Farbgebung im AIORES-Markenstil und eine eindeutige Absender- und Betreffstruktur. Ziel ist es, dass jede automatisiert versendete Nachricht denselben professionellen Eindruck hinterlässt wie ein individuell verfasstes Schreiben – unabhängig davon, ob es sich um eine Erinnerung, eine Auftragsbestätigung oder eine Fristmitteilung handelt.

Dieser Anspruch gilt ausdrücklich auch für die automatisch generierten PDF-Dokumente, die den E-Mails als Anlage beigefügt werden – etwa Protokolle, Bestätigungen, Auftrags- oder Erinnerungsdokumente. Diese Anhänge werden bewusst auf zwei, auf den ersten Blick gegensätzliche Ziele hin optimiert: eine hochwertige, gestochen scharfe Optik mit konsistentem Corporate Design einerseits, und eine möglichst geringe Dateigröße andererseits. Eine sorgfältig abgestimmte Erzeugung von Schrift, Vektorgrafik und Logo-Elementen sowie eine komprimierte, aber verlustarme Einbindung von Bildmaterial stellen sicher, dass die Anlagen auch bei großem Empfängerkreis schnell zugestellt werden, ohne Postfächer unnötig zu belasten oder Qualitätseinbußen in der Darstellung hinzunehmen.

3.4 Dokumentengenerierung als durchgängiges Prinzip

Module mit Dokumentenausgabe (Protokolle, Abrechnungen, Bestätigungen) folgen demselben Verarbeitungsmuster: Strukturierte Eingabedaten werden in eine Vorlage überführt, als PDF generiert und über das zentrale Zugriffssystem bereitgestellt beziehungsweise per E-Mail versendet. Dieses wiederkehrende Muster erleichtert eine spätere technische Vereinheitlichung der Dokumentengenerierung in einer eigenständigen Infrastruktur.

4. Technische Weiterverarbeitung der Inhalte

Die Weiterverarbeitung der in den Bausteinen erfassten Inhalte folgt einem klar nachvollziehbaren Ablauf, der unabhängig vom konkreten Modul in vergleichbarer Form auftritt:

1. Erfassung – Eingabe strukturierter Daten über Formulare innerhalb der jeweiligen WebApp-Oberfläche
2. Validierung – Prüfung auf Vollständigkeit und Plausibilität vor Weiterverarbeitung
3. Verarbeitung – Verknüpfung mit zentralen Stammdaten, Anwendung modulspezifischer Logik (z. B. Fristberechnung, Kategorisierung)
4. Dokumenten- und E-Mail-Erzeugung – Generierung von PDF-Dokumenten und E-Mail-Inhalten auf Basis einheitlicher, markenkonformer Vorlagen mit optimierter Dateigröße
5. Bereitstellung – Zustellung per E-Mail (inklusive Anhang) oder Bereitstellung über das zentrale Zugriffssystem
6. Protokollierung – Ablage des Vorgangs im zentralen Journal zur lückenlosen Nachvollziehbarkeit

4.1 Automatisierte Dokumentenverarbeitung (Rechnungswesen)

Im Bereich der Rechnungsverarbeitung erfolgt die Weiterverarbeitung mehrstufig: Nach automatischer Texterkennung werden Rechnungen anhand eines Regelwerks Objekten, Dienstleistern und Kontenplänen zugeordnet. Datensätze, die nicht eindeutig zugeordnet werden können, werden automatisch in einen separaten Nachbearbeitungsbereich verschoben, gezielt manuell bearbeitet und anschließend mit dem Hauptbestand abgeglichen. Dieser Prozess wird begleitend in einem Excel-Bericht dokumentiert, der eine Qualitätskontrolle über alle verarbeiteten Vorgänge ermöglicht.

4.2 Geplante KI-gestützte Vorgangsbearbeitung

Konzeptionell vorgesehen ist ein auf der Microsoft-Graph-API basierendes System zur automatisierten Klassifizierung eingehender E-Mails sowie zur Erstellung von Antwortentwürfen. Eine automatische Versendung ohne menschliche Prüfung ist dabei explizit nicht vorgesehen. Voraussetzung für die Umsetzung ist eine noch ausstehende Azure-AD-App-Registrierung. Die Integration soll vorrangig über die bestehende WebApp-Familie erfolgen, um zusätzliche, separate API-Kosten zu vermeiden.

5. Strategischer Weg zur eigenständigen Servertechnik

Die aktuelle Architektur auf Basis von Google Apps Script ist eine bewusst gewählte Übergangslösung. Sie ermöglicht eine schnelle, kostengünstige Umsetzung einzelner Bausteine ohne eigene Serverinfrastruktur. Mit wachsendem Funktionsumfang, steigender Nutzerzahl und zunehmender Datenmenge stößt dieser Ansatz jedoch an technische und wirtschaftliche Grenzen – etwa hinsichtlich Ausführungszeiten, Skalierbarkeit, Performance bei komplexeren Abfragen und gestalterischer Flexibilität der Benutzeroberfläche.

5.1 Auslösekriterien für die Migration

Der Wechsel auf eigene Servertechnik ist nicht an einen festen Zeitpunkt gebunden, sondern an das tatsächliche Volumen und den Reifegrad der entwickelten Bausteine. Folgende Indikatoren sprechen für den Beginn der Migration:

- Eine kritische Anzahl produktiv genutzter Module mit stabiler, ausgereifter Funktionslogik
- Spürbare Performance- oder Skalierungsgrenzen innerhalb der bestehenden Plattform
- Wachsender Bedarf an individuellen Schnittstellen (APIs) zu Drittsystemen
- Anforderungen an Benutzeroberfläche oder mobile Nutzung, die mit der aktuellen Plattform nicht mehr abbildbar sind
- Ausreichendes Datenvolumen und Nutzerzahl, die eine dedizierte Datenbankarchitektur wirtschaftlich rechtfertigen

5.2 Vorbereitende Maßnahmen im aktuellen Entwicklungsstadium

Um eine spätere Migration möglichst reibungslos zu gestalten, werden bereits im aktuellen Entwicklungsstadium Grundsätze beachtet, die eine spätere Übertragbarkeit sicherstellen:

- Konsequente Trennung von Stammdaten und modulspezifischer Logik
- Vermeidung modul-individueller Datenstrukturen zugunsten einheitlicher, klar dokumentierter Schemata
- Plattformunabhängige Gestaltung von Zugriffs- und Authentifizierungskonzepten
- Verwendung relativer statt fest verdrahteter Pfade und Konfigurationswerte
- Dokumentation wiederkehrender Verarbeitungsmuster zur späteren Übertragung in eine einheitliche Backend-Logik, einschließlich der Vorlagen für E-Mail- und PDF-Generierung

5.3 Zielarchitektur (konzeptionell)

Die langfristige Zielarchitektur sieht eine eigenständige, serverbasierte Anwendung vor, bestehend aus einer zentralen Datenbank, einer modularen Backend-Logik je Funktionsbereich sowie einer einheitlichen, eigenständig gestaltbaren Benutzeroberfläche. Die heute in Google Apps Script abgebildeten Module sollen dabei nicht von Grund auf neu konzipiert, sondern in ihrer fachlichen Logik übernommen und technisch auf die neue Infrastruktur übertragen werden.

Bereich	Heute	Zielzustand
Infrastruktur	Google Apps Script	Eigener Server / Cloud-Hosting
Datenhaltung	Google Sheets	Relationale Datenbank
Authentifizierung	Token-basiert (Apps Script)	Eigenständiges Auth-System

Bereich	Heute	Zielzustand
Dokumentenausgabe	PDF via Apps Script	Zentrale Dokumenten-Engine
E-Mail-/PDF-Layout	Einheitliche Vorlagen, optimiert	Zentrales Template- & Rendering-System
Schnittstellen	Eingeschränkt	Offene API-Anbindung an Drittsysteme

6. Zusammenfassung

AIORES befindet sich in einer kontrollierten, iterativen Entwicklungsphase. Der gewählte Ansatz – Entwicklung einzelner, praxiserprobter Bausteine auf einer kosteneffizienten, schnell verfügbaren Plattform – ermöglicht es, Funktionsumfang und Nutzen frühzeitig zu validieren, bevor in eine eigenständige technische Infrastruktur investiert wird.

Bereits im aktuellen Entwicklungsstadium wird besonderer Wert auf ein hochwertiges, markenkonformes Erscheinungsbild der gesamten Kommunikation gelegt – von der E-Mail-Gestaltung bis zu den automatisch erzeugten PDF-Anhängen, die bei minimaler Dateigröße eine gestochen scharfe, professionelle Optik bieten.

Der Übergang auf eigene Servertechnik ist als bewusster, am tatsächlichen Entwicklungsstand orientierter Schritt angelegt und nicht als vorab fixierter Zeitpunkt. Durch die konsequente Beachtung übertragbarer Architekturprinzipien bereits in der aktuellen Phase wird sichergestellt, dass dieser Schritt vollzogen werden kann, sobald Volumen und Reifegrad der Software dies erfordern – ohne die bestehende fachliche Logik und die bereits gesammelte Praxiserfahrung zu verlieren.