



blocpress
document automation

Solution Design Concept

Vision

blocpress ist ein template-basierter Print-Server, der die automatisierte Erstellung professioneller Dokumente durch intelligente Zusammenführung von LibreOffice-Templates mit strukturierten Daten ermöglicht. Die Lösung transformiert den Dokumentenerstellungsprozess von manueller, fehleranfälliger Arbeit zu einem kontrollierten, qualitätsgesicherten und automatisierten Workflow.

blocpress unterstützt Compliance-Anforderungen durch einen turnusmäßigen Review-Prozess. Dokumente werden nach 1, 3 oder 5 Jahren automatisch zur Prüfung auf inhaltliche Korrektheit vorgelegt, um sicherzustellen, dass Templates stets den aktuellen rechtlichen und fachlichen Anforderungen entsprechen.

Die Qualitätssicherung wird durch umfassende Test-Unterstützung gewährleistet. Abnehmende können Testdaten ergänzen und duplizieren, automatisch PDFs zur Abnahme generieren lassen und diese in einem Testpool für Regressionstests hinterlegen. Dies ermöglicht eine kontinuierliche Validierung der Dokumentengenerierung auch bei Template-Änderungen.

Eine intelligente Suchfunktion ermöglicht die Recherche nach fachlichen Konstrukten über alle Templates und Bausteine hinweg. Nutzer können ermitteln, in welchen Dokumenten spezifische Konzepte wie Beitragsanpassung, Deckungsumfang oder Vertragsklauseln verwendet werden, um Konsistenz zu gewährleisten und Änderungen koordiniert durchzuführen.

Technische Umsetzung der Vision: [Technische Ziele im System Design Concept](#)

Value Proposition

Kundensegment: Entwicklungsteams & Fachabteilungen

Entwicklungsteams und Fachabteilungen, die regelmäßig Dokumente mit variablen Inhalten erstellen müssen (z.B. Verträge, Berichte, Rechnungen, Zertifikate), erhalten eine Lösung, die:

- Die Dokumentenerstellung vollständig über APIs automatisiert

- Templates zentral verwaltet und versioniert
- Wiederholungsgruppen und bedingte Inhalte unterstützt
- Qualitätskontrolle und Freigabeprozesse integriert
- Als Docker-Container einfach in bestehende Infrastrukturen integrierbar ist

Kundensegment: Template-Designer & Qualitätsmanager

Personen, die für die Erstellung und Pflege von Dokumentvorlagen sowie die Qualitätssicherung verantwortlich sind, profitieren von:

- Einer vertrauten LibreOffice-Umgebung zur Template-Erstellung
- Wiederverwendbaren Bausteinen für konsistente Inhalte
- Einem strukturierten Freigabeprozess mit Rollen und Berechtigungen
- Qualitätskontroll-Mechanismen vor der finalen Freigabe

Value Creation Architecture

Die Value Creation Architecture beschreibt die Struktur der Lösung zur Wertschöpfung:

Kunden & Nutzer

- Entwicklungsteams: Nutzen die REST-API zur automatisierten Dokumentengenerierung aus Anwendungen heraus
- Template-Designer: Erstellen und pflegen LibreOffice-Templates mit User-Fields und Strukturelementen
- Qualitätsmanager: Prüfen generierte Dokumente und geben Templates frei
- Administratoren: Verwalten Berechtigungen und überwachen den Freigabeprozess
- Compliance-Reviewer: Führen turnusmäßige Reviews durch und prüfen Templates auf inhaltliche Aktualität
- Test-Manager: Pflegen Testdaten, führen Abnahmetests durch und verwalten den Regressionstestpool

Rollen und deren Systemzugang: *User Types im System Design Concept*

Organisationen

- Kundenorganisation: Betreibt blocpress als Teil ihrer Infrastruktur
- Fachabteilungen: Definieren inhaltliche Anforderungen an Templates und Dokumente

Digitale Elemente

- blocpress (Docker-Container): Kernkomponente zur Dokumentengenerierung und Template-Verwaltung
- REST-API: Schnittstelle für Template-Upload, Dokumentengenerierung, Workflow-Steuerung und Suche
- LibreOffice-Engine: Verarbeitet Templates und führt Daten zusammen
- Template-Repository: Speichert Templates, Bausteine und deren Versionen
- Workflow-Engine: Steuert Qualitätskontrolle und Freigabeprozess
- Anwendungssysteme: Konsumieren die API zur Dokumentengenerierung
- Compliance-Manager: Überwacht Review-Zyklen, erstellt Benachrichtigungen für anstehende Reviews und verwaltet Review-Historie
- Test-Framework: Verwaltet Testdaten, generiert Abnahme-PDFs, führt Regressionstests durch und speichert Testbaselines
- Content Search Engine: Indexiert Templates und Bausteine und ermöglicht die Recherche nach fachlichen Konstrukten über alle Dokumente hinweg

Technische Beschreibung der Software-Elemente: [Software Elements im System Design Concept](#)

Information Architecture

Die Informationsarchitektur definiert die zentralen Geschäftsentitäten. *Technische Umsetzung als Datenbank-Entities:* [Entities im Element Design Concept](#)

Template

Ein Template ist ein LibreOffice-Dokument mit definierten Variablen und Strukturelementen. Attribute: Template-ID, Name, Beschreibung, Version, Status (Entwurf, In Prüfung, Freigegeben, Archiviert), Ersteller, Erstellungsdatum, Freigeber, Freigabedatum, User-Fields-Definition, Wiederholungsgruppen-Definition, Bedingungen.

Entity-Definition: [E-1: Template](#)

Baustein

Ein Baustein ist ein wiederverwendbares LibreOffice-Fragment, das in mehreren Templates eingebunden werden kann. Attribute: Baustein-ID, Name, Beschreibung, Typ (Kopfzeile, Fußzeile, Klausel, Abschnitt), Inhalt, Version, verwendende Templates.

Entity-Definition: [E-2: Baustein](#)

User-Field

Ein User-Field definiert einen Platzhalter im Template, der durch JSON-Daten befüllt wird. Attribute: Field-Name (in Punkt-Notation, z.B. `kunde.name`), Datentyp (String, Number, Date, Boolean), Formatierung, Standardwert, Pflichtfeld, Beschreibung.

Wiederholungsgruppe

Eine Wiederholungsgruppe definiert einen Bereich im Template (Section oder Table), der für jedes Element eines JSON-Arrays wiederholt wird. Attribute: Gruppen-ID, Typ (Section, Table), Array-Referenz (JSON-Path), enthaltene User-Fields.

Dokumentengenerierung

Eine Dokumentengenerierung ist ein Prozess, bei dem ein Template mit JSON-Daten befüllt wird. Attribute: Generierungs-ID, Template-ID mit Version, JSON-Eingabedaten, generiertes Dokument, Zeitstempel, Status (In Bearbeitung, Erfolgreich, Fehlgeschlagen), Fehlerprotokoll, Requestor.

Entity-Definition: [E-3: Dokumentengenerierung](#)

Freigabeprozess

Ein Freigabeprozess definiert die Schritte zur Qualitätssicherung eines Templates. Attribute: Prozess-ID, Template-ID, Workflow-Status (Eingereicht, In Prüfung, Rückfragen, Freigegeben, Abgelehnt), Prüfer, Prüfkomentare, Prüfdatum, Freigeber, Freigabedatum.

Entity-Definition: [E-4: Freigabeprozess](#)

Benutzer & Rolle

Benutzer und deren Rollen steuern Zugriffsrechte. Attribute: Benutzer-ID, Name, E-Mail, Rollen (Template-Designer, Qualitätsprüfer, Freigeber, Administrator, Compliance-Reviewer, Test-Manager), Berechtigungen (Template erstellen, Template bearbeiten, Template prüfen, Template freigeben, API nutzen, Review durchführen, Tests verwalten).

Entity-Definition: [E-9: Benutzer](#)

Compliance Review

Ein Compliance Review ist eine turnusmäßige Prüfung eines Templates auf inhaltliche Aktualität. Attribute: Review-ID, Template-ID, Review-Typ (1-Jahres-Review, 3-Jahres-Review, 5-Jahres-Review), Fälligkeitsdatum, Review-Status (Anstehend, In Bearbeitung, Abgeschlossen, Überfällig), Reviewer, Review-Datum, Review-Ergebnis (Freigegeben, Überarbeitung erforderlich, Archivieren), Kommentare, nächstes Review-Datum.

Entity-Definition: [E-5: Compliance Review](#)

Test Case

Ein Test Case definiert einen Testfall für ein Template mit spezifischen Testdaten. Attribute: Test-ID, Template-ID mit Version, Test-Name, Beschreibung, JSON-Testdaten, erwartetes PDF-Ergebnis (Baseline), aktuelles PDF-Ergebnis, Test-Status (Bestanden, Fehlgeschlagen, Baseline fehlt), Ersteller, Erstellungsdatum, letzte Ausführung, Test-Typ (Abnahmetest, Regressionstest).

Entity-Definition: [E-6: Test Case](#)

Testpool

Ein Testpool ist eine Sammlung von Test Cases für Regressionstests. Attribute: Pool-ID, Name, Beschreibung, enthaltene Test Cases, letzte Ausführung, Erfolgsrate, Template-IDs (für welche Templates der Pool gilt).

Entity-Definition: [E-7: Testpool](#)

Content Index

Der Content Index ermöglicht die Recherche nach fachlichen Konstrukten über alle Templates und Bausteine hinweg. Attribute: Index-ID, Suchbegriff (z.B. "Beitragsanpassung", "Deckungsumfang"), Vorkommen (Liste von Template-IDs und Baustein-IDs mit Positionsangaben), Kontext (umgebender Text), Kategorie (fachliches Konzept, rechtliche Klausel, technischer Begriff), letzte Aktualisierung.

Entity-Definition: [E-8: Content Index](#)

Business Processes

Template erstellen und freigeben

- Ein Template-Designer erstellt in LibreOffice ein Dokument mit User-Fields, die JSON-Pfade in Punkt-Notation referenzieren (z.B. `vertrag.kunde.name`)
- Der Designer markiert Sections oder Tables als Wiederholungsgruppen, indem er auf JSON-Arrays verweist
- Der Designer fügt IF-Bedingungen ein, um Inhalte basierend auf Datenwerten ein- oder auszublenden
- Das fertige Template wird über die Web-Oberfläche von blocpress hochgeladen
- blocpress validiert das Template auf Korrektheit der Field-Definitionen
- Der Designer reicht das Template zur Freigabe ein
- Ein Qualitätsprüfer überprüft das Template mit Testdaten und kommentiert eventuelle Probleme
- Bei Rückfragen bearbeitet der Designer das Template und reicht es erneut ein

- Ein Freigeber gibt das Template final frei, sodass es für die Produktivnutzung verfügbar ist

Use Cases: *UC-1: Template hochladen*, *UC-2: Zur Freigabe einreichen*, *UC-8: Template freigeben*

Systemszenario: *Template hochladen und zur Freigabe einreichen*

Dokument automatisiert generieren

- Eine Anwendung sendet einen API-Request mit Template-ID und JSON-Daten an blocpress
- blocpress lädt das freigegebene Template aus dem Repository
- Die LibreOffice-Engine befüllt alle User-Fields mit Werten aus den JSON-Daten gemäß der Punkt-Notation
- Wiederholungsgruppen werden für jedes Element des referenzierten JSON-Arrays dupliziert und befüllt
- IF-Bedingungen werden ausgewertet und entsprechende Inhalte ein- oder ausgeblendet
- Das generierte Dokument wird als ODT, PDF oder RTF exportiert
- Die Anwendung erhält das fertige Dokument als Download-Link oder Base64-String

Technical Function: *TF-5: Dokument generieren*

Use Case: *UC-10: Testdokument mit Template generieren*

Systemszenario: *Dokument über API generieren*

Bausteine verwalten und wiederverwenden

- Ein Template-Designer erstellt einen wiederverwendbaren Baustein (z.B. Standardfußzeile, AGBs)
- Der Baustein wird in blocpress als eigenständiges Element gespeichert
- Bei der Template-Erstellung kann der Designer Bausteine aus einer Bibliothek auswählen und einbinden
- Änderungen am Baustein werden zentral vorgenommen
- Alle Templates, die den Baustein verwenden, profitieren automatisch von der Aktualisierung (nach erneutem Freigabeprozess)

Turnusmäßiges Compliance Review durchführen

- Der Compliance-Manager überwacht alle Templates und erstellt automatisch Review-Aufgaben basierend auf dem definierten Review-Zyklus (1, 3 oder 5 Jahre)
- Ein Compliance-Reviewer erhält eine Benachrichtigung über anstehende Reviews
- Der Reviewer öffnet das Template und prüft die inhaltliche Korrektheit gegen aktuelle rechtliche und fachliche Anforderungen
- Bei festgestellten Mängeln dokumentiert der Reviewer diese und markiert das Template zur Überarbeitung
- Ein Template-Designer bearbeitet das Template entsprechend den Review-Kommentaren

- Das überarbeitete Template durchläuft den regulären Freigabeprozess
- Bei erfolgreicher Freigabe schließt der Reviewer das Review ab und das System plant automatisch das nächste Review
- Veraltete Templates können vom Reviewer zur Archivierung vorgeschlagen werden

Technical Function: [TF-7: Compliance-Reviews überprüfen](#)

Use Case: [UC-12: Review durchführen](#)

Systemszenario: [Turnusmäßiges Compliance Review](#)

Template testen und Regressionstests durchführen

- Ein Test-Manager erstellt einen Test Case mit JSON-Testdaten für ein Template
- Der Test-Manager kann bestehende Testdaten duplizieren und anpassen, um verschiedene Szenarien abzudecken
- Das Test-Framework generiert automatisch ein PDF aus dem Template und den Testdaten
- Der Abnehmende prüft das generierte PDF auf Korrektheit
- Bei erfolgreicher Abnahme wird das PDF als Baseline (erwartetes Ergebnis) im Test Case hinterlegt
- Der Test Case wird einem Testpool für Regressionstests hinzugefügt
- Bei jeder Template-Änderung führt das Test-Framework automatisch alle Tests aus dem zugehörigen Testpool aus
- Das System vergleicht die generierten PDFs mit den gespeicherten Baselines
- Abweichungen werden dem Test-Manager gemeldet, der entscheidet, ob es sich um einen Fehler oder eine gewünschte Änderung handelt
- Bei gewünschten Änderungen aktualisiert der Test-Manager die Baseline mit dem neuen PDF

Technical Functions: [TF-5: Dokument generieren](#), [TF-6: PDF vergleichen](#)

Use Cases: [UC-14: Test Case erstellen](#), [UC-16: Test ausführen](#)

Systemszenario: [Automatischer Regressionstest](#)

Fachliche Konstrukte recherchieren

- Ein Fachexperte oder Template-Designer benötigt eine Übersicht, in welchen Dokumenten ein spezifisches fachliches Konstrukt verwendet wird (z.B. "Beitragsanpassung", "Deckung", "Widerrufsrecht")
- Der Nutzer gibt den Suchbegriff in die Suchfunktion von blocpress ein
- Die Content Search Engine durchsucht alle Templates und Bausteine nach dem Begriff
- Das System zeigt eine Liste aller Templates und Bausteine, in denen der Begriff vorkommt, mit Kontext und Positionsangabe
- Der Nutzer kann die Ergebnisse nach Template-Typ, Baustein-Kategorie oder Verwendungshäufigkeit filtern

- Bei Auswahl eines Suchergebnisses öffnet blocpress das entsprechende Template oder den Baustein an der relevanten Stelle
- Der Nutzer kann eine Änderung am fachlichen Konstrukt vornehmen und sieht direkt, welche anderen Templates ebenfalls betroffen sind
- Änderungen in Bausteinen werden automatisch in allen verwendenden Templates berücksichtigt und der Suchindex wird aktualisiert

Use Case: [UC-19: Nach Begriff suchen](#)

Systemszenario: [Fachliche Konstrukte über alle Templates suchen](#)

Quality Requirements

- Verfügbarkeit: blocpress soll eine Verfügbarkeit von 99.5% während Geschäftszeiten erreichen
- Performance: Die Generierung eines Standard-Dokuments (bis 20 Seiten) soll innerhalb von 5 Sekunden abgeschlossen sein
- Skalierbarkeit: Der Server soll bis zu 100 gleichzeitige Dokumentengenerierungen verarbeiten können
- Wartbarkeit: Templates und Bausteine sollen versioniert werden, sodass Änderungen nachvollziehbar sind
- Sicherheit: Der Zugriff auf Templates und die API soll durch rollenbasierte Authentifizierung geschützt sein
- Benutzerfreundlichkeit: Template-Designer sollen ohne technische Schulung Templates in LibreOffice erstellen können
- Zuverlässigkeit: Die Dokumentengenerierung soll deterministisch sein – gleiche Eingabedaten erzeugen immer identische Dokumente
- Compliance: Templates mit rechtlicher oder regulatorischer Relevanz müssen automatisch nach z.B. 1, 3 oder 5 Jahren zur Review-Prüfung vorgelegt werden
- Testbarkeit: Jedes Template muss mit Testdaten validierbar sein, und Regressionstests müssen innerhalb von 30 Sekunden pro Test Case abgeschlossen sein
- Nachvollziehbarkeit: Alle Compliance-Reviews, Testausführungen und deren Ergebnisse müssen für Audits protokolliert und abrufbar sein
- Suchgeschwindigkeit: Die Suche nach fachlichen Konstrukten über alle Templates und Bausteine soll innerhalb von 2 Sekunden Ergebnisse liefern

Technische Qualitätsanforderungen: [Quality Requirements im System Design Concept](#)

Element-spezifische Qualitätsanforderungen: [Quality Requirements im Element Design Concept](#)

Constraints

- Technologische Beschränkung: blocpress muss als Docker-Container betrieben werden können
- Datenschutz: Verarbeitung personenbezogener Daten in Templates muss DSGVO-konform erfolgen

- Format-Beschränkung: Nur LibreOffice-kompatible Formate (ODT, ODF) und Export nach ODT/PDF/RTF werden unterstützt
- Datenformat: Eingabedaten müssen als valides JSON bereitgestellt werden
- Organisatorisch: Der Freigabeprozess muss eine Vier-Augen-Prüfung gewährleisten (Designer ≠ Freigeber)
- Lizenzierung: Verwendung von Open-Source-Komponenten (LibreOffice) unter Einhaltung der jeweiligen Lizenzbedingungen
- Compliance-Nachweis: Review-Zyklen und deren Dokumentation müssen branchenspezifischen Compliance-Anforderungen entsprechen (z.B. ISO, Branchenstandards)

Technische Constraints: [Constraints im System Design Concept](#)

Element-spezifische Constraints: [Constraints im Element Design Concept](#)