Prozessdeck

WHITEPAPER · MAI 2026

Funktion und Architektur.

Eine technische Übersicht der Plattform: Konzepte, vollständiger Funktionsumfang, Sicherheitsmechanismen und Architektur — von der Datenstruktur bis zum Audit-Log.

Dokument: Technisches Whitepaper · **Domain:** prozessdeck.de

v2.2 · 2026-05-06

INHALT

Was dich in diesem Dokument erwartet.

Dieses Whitepaper beschreibt Prozessdeck vollständig — von der konzeptionellen Idee bis hinunter zu jeder einzelnen Sicherheitsmaßnahme im Code. Sieben Kapitel, kompakt aufgebaut, ohne Marketing-Sprache.

IN ZAHLEN

Prozessdeck umfasst **40 Tabellen** im Schema, **20 Routen-Module** im Backend, **21 Services**, **11 Funktionsmodule** und ein vollständiges Sicherheitsmodell, das in **vier Audit-Pässen** mit insgesamt **76+ identifizierten Findings** reifegeprüft wurde.

01 · PLATTFORM-ÜBERSICHT

Was Prozessdeck ist.

Prozessdeck ist eine webbasierte Plattform zur strukturierten Abbildung wiederkehrender Geschäftsprozesse. Vorlagen werden einmal definiert, beliebig oft als isolierte Instanzen ausgeführt und vollständig protokolliert. Das System ist als Single-Page-Application konzipiert, läuft im Browser und benötigt keine clientseitige Installation.

TABELLEN IM SCHEMA **40**	ROUTES / BLUEPRINTS **20**	PERMISSION-KEYS **13**	SERVER-STANDORT **EU · Frankfurt**
FUNKTIONSMODULE **11**	BACKGROUND-SERVICES **21**	AUDIT-PÄSSE ABGESCHLOSSEN **4**	BEHOBENE FINDINGS **76+**

Was die Plattform leistet

Prozessdeck wandelt unstrukturierte, in Office-Werkzeugen verstreute Abläufe in strukturierte, nachvollziehbare und auditierbare Prozesse um. Drei Eigenschaften definieren das System:

KLARHEIT

Jeder Schritt hat einen Verantwortlichen, einen Status, eine Frist. Keine Mehrdeutigkeit, keine Versionskonflikte, keine doppelten Pflegeorte. Wer auf einen Run schaut, sieht in Sekunden den Zustand.

VERBINDLICHKEIT

Schritte werden über konfigurierbare Confirmation-Mechanismen freigegeben. Pflicht-Freitexte, Namens-Signoffs und Datei-Signaturen sind erzwingbar — ein Schritt kann ohne die geforderte Bestätigung nicht abgeschlossen werden.

NACHVOLLZIEHBARKEIT

Jede sicherheitsrelevante Aktion wird in einem dedizierten Audit-Log gespeichert. Aktor, Aktion, Objekt, Zeitstempel — lückenlos und exportierbar. Für interne Reviews wie für externe Prüfer.

Eckdaten auf einen Blick

Plattform-Typ	Webbasierte Single-Page-Application mit serverseitiger Datenhaltung
Architektur	Multi-Tenant per Container-Isolation — jede Organisation erhält eine eigene Instanz mit dedizierter Datenbank und eigenem Volume
Tech-Stack	Python 3 + Flask 3, SQLite (WAL), Vanilla JavaScript (ES-Module, ohne Build-Step)
Authentifizierung	Session-basiert mit URL-safen Token (32 Byte), optionale Zwei-Faktor-Authentifizierung (TOTP nach RFC-6238), Backup-Codes
Datenhaltung	SQLite-Datenbank pro Tenant, Datei-Anhänge auf gemountetem Volume, sensible Felder zusätzlich verschlüsselt
Compliance	DSGVO-konform, lückenloses Audit-Log, exportierbare Daten
Hosting	EU-Rechenzentrum (Hostinger Frankfurt), deutsche Datenhoheit, Auftragsverarbeitungsvertrag mit Hosting-Anbieter
Mobile-Unterstützung	Responsive für alle Geräteklassen, Touch-Targets ≥ 44 px, Off-Canvas-Drawer-Navigation, Agenda-View für Kalender

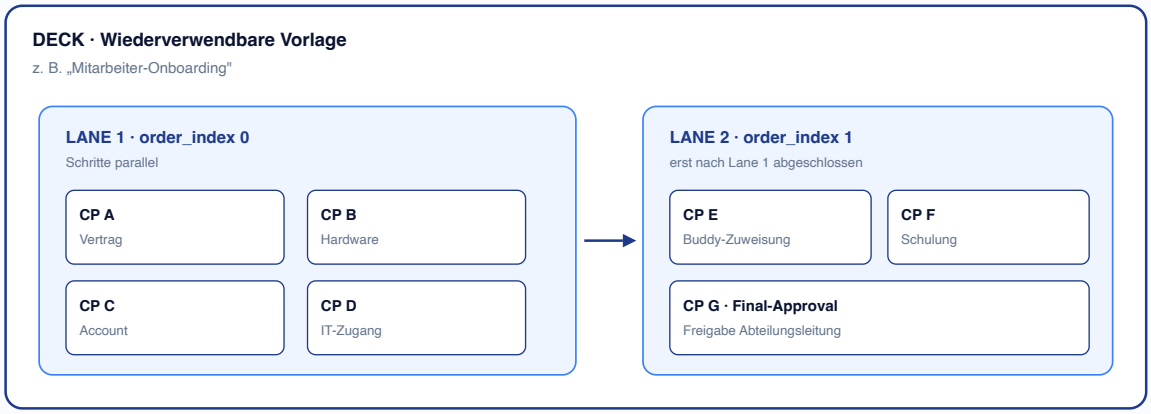
> **DESIGNPRINZIP**
>
> Prozessdeck folgt dem Grundsatz „so viel Struktur wie nötig, so wenig Software wie möglich". Kein Build-Step, kein JavaScript-Framework, keine Datenbank-Server-Konfiguration. Was im Repo liegt, läuft im Browser — debugbar, lesbar, in fünf Jahren noch wartbar.

02 · KONZEPTE UND DATENMODELL

Die vier Grundbegriffe.

Das Datenmodell von Prozessdeck baut auf vier Kernbegriffen auf. Jeder dieser Begriffe entspricht einer eigenen Tabelle im Schema und einer eigenen Sicht in der Anwendung.

HIERARCHIE · DECK → LANE → CHECKPOINT → RUN

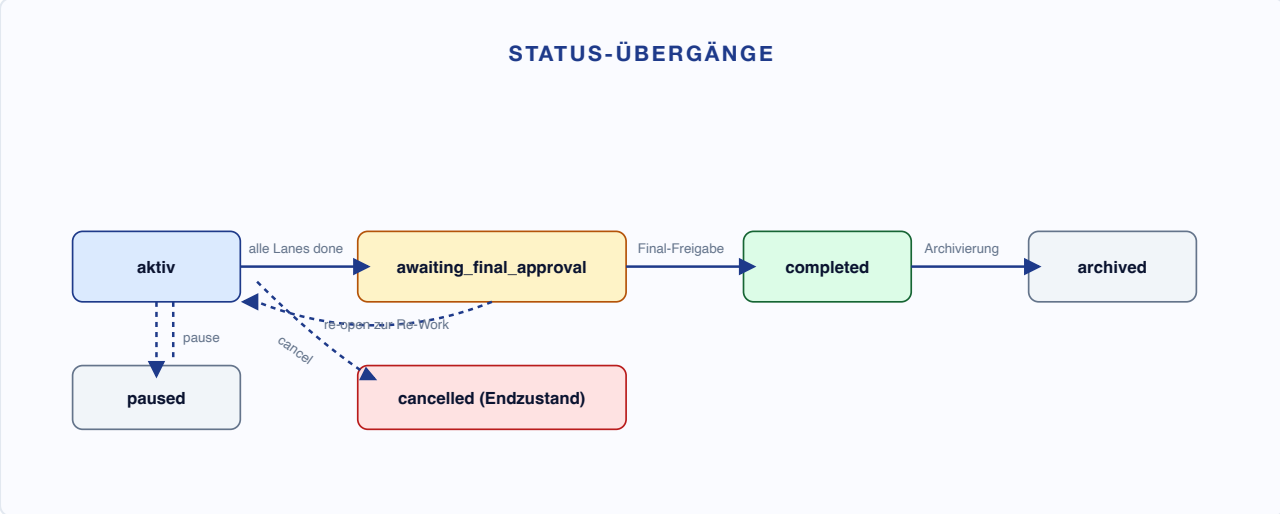


Kernbegriffe

Deck	Wiederverwendbare Vorlage eines Prozesses mit beliebig vielen Schritten (Checkpoints), Conditions, Verantwortlichkeiten und Fristen. Decks sind versioniert — jede Veröffentlichung erzeugt eine eigene Version mit Diff-Vergleich zur Vorgängerversion.
Run	Konkrete Ausführung eines Decks. Beim Start wird ein Snapshot erstellt — spätere Deck-Änderungen beeinflussen laufende Runs nicht. Ein Run kennt einen Lifecycle mit definierten Zuständen und Übergängen.
Lane	Gruppe paralleler Schritte (gleicher `order_index`). Lanes werden sequenziell durchlaufen, innerhalb einer Lane laufen die Schritte parallel. So lassen sich Phasen mit Vorgänger-/Nachfolger-Logik abbilden, ohne starre Linearität zu erzwingen.
Checkpoint	Einzelner Schritt mit konfigurierbarem Confirmation-Typ, optionaler Frist (SLA), optionaler Datei-Anforderung und optionalen Zuweisungen. Checkpoints können an Bedingungen geknüpft werden, die sich auf vorherige Schritt-Ergebnisse beziehen.

Run-Lifecycle

Ein Run durchläuft definierte Zustände, die im Datenmodell abgebildet sind. Jeder Übergang erzeugt einen Eintrag im `run_events`-Log und — bei sicherheitsrelevanten Wechseln — zusätzlich im `audit_log`.



Zusätzliche Übergänge: **aktiv → paused** (alle SLAs werden eingefroren), **aktiv → cancelled** (Endzustand, kein Re-Open mehr), **awaiting_final_approval → aktiv** bei Ablehnung der Final-Freigabe (für Re-Work).

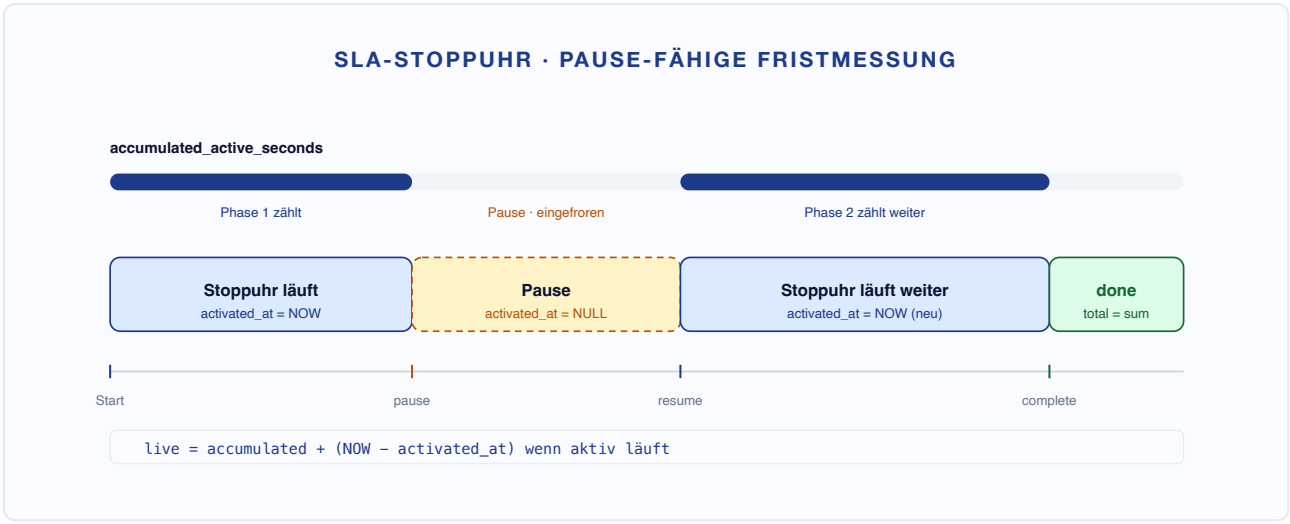
Activation-Engine

Die Engine arbeitet lane-basiert. Beim Run-Tick (ausgelöst durch Schritt-Bestätigung, Run-Start, Run-Resume oder periodisch durch den SLA-Worker, Default-Intervall 15 Minuten, konfigurierbar via `SLA_CHECK_INTERVAL_MINUTES`) prüft sie drei Stufen:

01 · Lane-Sequenz	Die Engine durchläuft Lanes nach `order_index` aufsteigend. Eine neue Lane wird erst aktiviert, wenn alle Schritte der vorherigen Lane den Status `done` oder `skipped` haben.
02 · Schritt-Aktivierung	Innerhalb der aktuellen Lane werden alle Pending-Schritte geprüft. Schritte ohne Bedingung oder mit erfüllter Bedingung werden aktiviert, ihre SLA-Stoppuhr gestartet und Zuweisungen benachrichtigt. Schritte mit unerfüllten Bedingungen werden übersprungen (`skipped`).
03 · Final-Approval	Wenn alle Lanes den Status `done` oder `skipped` erreicht haben, wechselt der Run in `awaiting_final_approval` und wartet auf die Freigabe durch den Approver.

SLA-Stoppuhr-Logik

Jeder Checkpoint kann eine Frist definieren. Die Stoppuhr läuft pro Schritt und ist pause-fähig — wichtig, wenn ein Run pausiert wird oder ein Schritt zur Wiederbearbeitung zurückgegeben wird.



- ✓ **Start:** `sla_activated_at = NOW` bei Schritt-Aktivierung
- ✓ **Pause** (Run pausiert oder Schritt abgelehnt): akkumulierte Zeit wird in `sla_accumulated_active_seconds` gespeichert, `sla_activated_at` wird auf `NULL` gesetzt
- ✓ **Resume:** `sla_activated_at = NOW` wird neu gesetzt, akkumulierte Zeit bleibt erhalten
- ✓ **Live-Berechnung:** `accumulated + (NOW − activated_at)` wenn aktiv läuft

Datenmodell — 40 Tabellen, vier Domänen

Prozess-Kern

`decks` `deck_checkpoints` `deck_checkpoint_files` `deck_checkpoint_subtasks` `deck_checkpoint_archive_links` `deck_versions` `deck_shares` `deck_final_approvers` `runs` `run_checkpoints` `run_checkpoint_assignees` `run_final_approvers` `run_events` `run_schedules` `checkpoint_comments` `checkpoint_todos` `checkpoint_files` `user_pinned_decks`

Account, Auth & Rollen

`users` `sessions` `pending_totp_challenges` `password_reset_tokens` `roles` `role_permissions` `departments` `user_departments`

Sub-Module

`calendar_accounts` `calendar_events` `meetings` `meeting_attendees` `notifications` `notification_settings` `audit_log` `signatures` `file_signature_fields` `archive_files` `archive_folders` `archive_folder_shares`

System

`app_settings` `settings`

03 · FUNKTIONSUMFANG IM DETAIL

Alle Funktionen, in Modulen gegliedert.

Die Plattform ist in elf logische Module unterteilt. Jedes Modul deckt einen klar abgegrenzten Anwendungsbereich ab und ist über die Hauptnavigation oder direkten Zugriff per URL erreichbar. Die folgende Auflistung ist vollständig — was nicht hier steht, ist nicht im Produkt.

01
Decks
Wiederverwendbare Vorlagen

02
Runs
Konkrete Ausführungen

03
Checkpoints
Einzelne Prozessschritte

04
Sub-Prozesse
Verschachtelte Workflows

05
Permissions
Rollen & Rechte

06
Kalender
Termine & Fristen

07
Statistik
KPIs & Trends

08
Team
User & Abteilungen

09
Audit-Log
Sicherheits-Events

10
Notifications
In-App & Mail

11
Sonstige
Querschnittsfunktionen

TOTAL
11
Funktionsmodule

MODUL 01

Decks — Vorlagen

Der Editor zur Definition wiederverwendbarer Prozess-Templates. Lane-basiert, versioniert, mit granularen Freigaben.

- ✓ Editor für Lanes und Schritte — Lanes als Ordnungsindex, Schritte mit Position innerhalb der Lane
- ✓ Drei Confirmation-Typen pro Schritt: `simple_ok` (einfaches OK), `yes_no` (Ja/Nein), `accept_reject` (Akzeptieren — Reject läuft über separaten Endpoint)
- ✓ Sub-Process-Schritte: ein Schritt kann ein anderes Deck als Sub-Run starten (über `sub_deck_id`)
- ✓ Conditional Activation: ein Schritt aktiviert sich nur, wenn ein vorgelagerter Schritt einen bestimmten Wert hat (`condition_checkpoint_id` + `condition_value`)
- ✓ Datei-Anforderungen pro Schritt — mit oder ohne Pflichtsignatur, mit definierbaren Signatur-Feldern
- ✓ Default-Assignees pro Schritt (User-IDs als JSON-Array)
- ✓ Final-Approver pro Deck — mit Mode `any` (einer reicht) oder `all` (alle erforderlich)
- ✓ Sharing-Mechanismus mit drei Stufen: Viewer / Editor / Admin (Tabelle `deck_shares`)
- ✓ Versionierung mit Diff-Vergleich zwischen beliebigen Versionen
- ✓ Zwei Sichten in der Deck-Master-Liste: Grid und Liste
- ✓ Export & Import als JSON für Migration und Backups
- ✓ Archive-Toggle mit Sichtbarkeitssteuerung (Status `active` / `archived`)

MODUL 02

Runs — Instanzen

Jede Ausführung eines Decks ist ein Run mit eigener Timeline, eigenem Status und eigener Historie. Snapshot-basiert — laufende Runs sind immun gegen nachträgliche Deck-Änderungen.

- ✓ Run-Liste mit vier Filter-Tabs: **Aktiv**, **Fertig**, **Abgebrochen**, **Archiv**
- ✓ Timeline-Detailansicht mit allen Lifecycle-Events (Tabelle `run_events`)
- ✓ Lane-Ansicht mit Live-Status pro Schritt: pending, active, completed, skipped, rejected
- ✓ Control-Buttons: Start, Pause, Resume, Cancel, Archive — jeweils mit Audit-Log-Eintrag
- ✓ Final-Approval-Mechanismus: Run wartet vor Abschluss auf Freigabe in `awaiting_final_approval`; bei Reject zurück nach `active` für Re-Work
- ✓ Multi-Final-Approver mit Mode `any` oder `all` (vom Deck eingefroren beim Start)
- ✓ Bulk-Aktionen für Massen-Operationen — Multi-Select bis maximal 100 Runs pro Aufruf
- ✓ Volltext-Suche über alle Run-Inhalte
- ✓ Run-Klonen mit CP-Reset auf pending
- ✓ Snapshot-basiert: Beim Run-Start wird die Deck-Version eingefroren, spätere Deck-Änderungen wirken sich nicht auf laufende Runs aus
- ✓ SLA-Frist-Tracking pro Schritt mit pause-fähiger Stoppuhr (siehe Sektion 02)

MODUL 03

Checkpoints — Schritte

Schritte sind die kleinste handlungsorientierte Einheit. Jeder Checkpoint kennt einen Confirmation-Typ, optional eine Frist, optional eine Datei-Anforderung.

- ✓ Confirm-Endpoints für alle drei Confirmation-Typen
- ✓ Reject-Endpoint mit optionalem Reject-Grund (Spalte `rejection_reason`)
- ✓ Reopen-Endpoint zur Wiederaufnahme bestätigter Schritte
- ✓ Reassign — Übergabe an andere Verantwortliche mit Notification
- ✓ Kommentar-Threads pro Checkpoint mit Inline-Editierbarkeit (Author oder Admin darf löschen)
- ✓ Sub-Todos pro Checkpoint mit Done/Pending-Status
- ✓ Datei-Uploads mit Drag-&-Drop, MIME-Type-Validierung und 50-MB-Größenlimit pro Datei
- ✓ Datei-Signaturen — mehrfach pro Datei möglich mit definierbaren Signatur-Feldern
- ✓ Optionaler PIN-Schutz für Signaturen (PIN min. 8, max. 32 Zeichen, Pbkdf2-Hash)
- ✓ Conditional Activation: ein Schritt wird nur aktiviert, wenn ein vorgelagerter Schritt einen bestimmten Wert geliefert hat
- ✓ SLA-Stoppuhr pro Schritt mit Pre-SLA-Mahnung und Post-SLA-Eskalation an die Führungsebene

MODUL 04

Sub-Prozesse

Decks können andere Decks als Sub-Run starten. So lassen sich wiederkehrende Teil-Workflows als eigene, wiederverwendbare Bausteine kapseln.

- ✓ Ein Deck-Schritt kann ein anderes Deck als Sub-Run starten — gesteuert über die Spalte `sub_deck_id` in `deck_checkpoints`
- ✓ Hauptprozess wartet, bis Sub-Run abgeschlossen ist, dann automatische Fortsetzung der Parent-Lane
- ✓ Eltern-Kind-Verknüpfung über `parent_run_id` und `parent_run_checkpoint_id` in der `runs`-Tabelle
- ✓ Cycle-Detection im Save-Pfad: A → B → A wird bei Konfiguration verhindert (Topo-Walk)
- ✓ Runtime-Cap mit 8-stufiger Tiefenbegrenzung als zweite Verteidigungslinie
- ✓ Outcome-Propagation: das Ergebnis des Sub-Runs (completed / cancelled / final_rejected) bestimmt das Verhalten des Parent-Schritts
- ✓ Hierarchische Anzeige im Run-Detail (Eltern-/Kind-Beziehung sichtbar)

MODUL 05

Permissions & Rollen

Drei Hauptrollen mit unterschiedlichen Sichten, plus Sharing-Modell pro Deck und 13 Permission-Keys mit Scope-Granularität.

- ✓ **Admin** — sieht und verwaltet alles, inklusive User-Management und Systemeinstellungen
- ✓ **Department-Lead** — sieht Decks, Runs und Mitarbeiter der eigenen Abteilung plus aller untergeordneten Abteilungen (rekursiv); kann Mitarbeiter aus dem gesamten Verantwortungsbereich zuweisen
- ✓ **Employee / User** — sieht eigene Decks, freigegebene Decks, Runs als Starter / Approver / Assignee

Pro Deck zusätzlich ein dreistufiges Sharing-Modell (Viewer / Editor / Admin) — granular kontrollierbar pro User oder Abteilung. 13 Permission-Keys mit Scope-Granularität (`own` / `department` / `all`) erlauben präzise Zugriffsregelungen.

MODUL 06

Kalender

Eigene Kalender-Sicht für alle Run-Ereignisse mit Mobile-Agenda und vorbereiteter Synchronisation.

- ✓ Monatsansicht mit allen Run-Ereignissen (Starts, Abschlüsse, Abbrüche, Fristen)
- ✓ Frist-Anzeige für offene Checkpoints
- ✓ Farbcodierung: Frist überzogen (rot), offen (gelb), Run gestartet (blau), abgeschlossen (grün), abgebrochen (grau)
- ✓ Mobile-Agenda-View statt 7-Spalten-Grid auf Smartphones
- ✓ Wochenansicht mit Zeitfenstern für Meetings
- ✓ Timezone-Support pro User
- ✓ CalDAV-Synchronisation vorbereitet (Nextcloud, Apple Calendar)
- ✓ Microsoft-Graph-Integration vorbereitet (Outlook 365)

MODUL 07

Statistik

Sieben Sichten auf das Prozess-Geschehen — von Bottlenecks bis Trends. Alles aus persistierten Metriken, kein Live-SQL-Hammer.

- ✓ **KPIs:** aktive Runs, abgeschlossene Runs, durchschnittliche Run-Dauer, SLA-Compliance-Rate
- ✓ **Bottlenecks:** häufigste Verzögerungs-Schritte (Top-N nach Wartezeit)
- ✓ **Workload-Verteilung:** wer arbeitet an wie vielen offenen Checkpoints
- ✓ **Top-Decks:** meistgenutzte Vorlagen mit Run-Volumen
- ✓ **Status-Verteilung:** Anteil aktiv / wartend / abgeschlossen
- ✓ **Eskalationen:** SLA-Überzüge der letzten 30 Tage
- ✓ **Trends:** Run-Volumen über Zeit (täglich / wöchentlich / monatlich)

MODUL 08

Team & Abteilungen

Vollständige User-Verwaltung mit Abteilungs-Strukturen und Bulk-Operationen.

- ✓ User-Verwaltung mit Rollen, Mail, Avatar, optionaler Telefonnummer
- ✓ Hierarchische Abteilungs-Strukturen mit Department-Leads — Parent-Lead sieht Sub-Abteilungen automatisch, Cycle-Schutz beim Verschachteln
- ✓ User-zu-Abteilung-Zuordnung (mehrere Abteilungen pro User möglich)
- ✓ Bulk-Aktionen für Massenanlage (Anlegen, Aktivieren, Deaktivieren)
- ✓ Signatur-PIN pro User (8 – 32 Zeichen, Pbkdf2-Hash) — Sperrung nach 5 fehlgeschlagenen Versuchen für 1 Stunde
- ✓ Avatar-Upload mit Image-Resizing und Auth-only-Auslieferung (keine Hotlink-Anfälligkeit)

MODUL 09

Audit-Log

Lückenloses, manipulationssicheres Protokoll aller sicherheitsrelevanten Ereignisse — getrennt vom funktionalen `run_events`-Log.

- ✓ Lückenlose Aufzeichnung aller Aktionen: Authentication-Events, User-Mutationen, Deck-/Run-/Checkpoint-Änderungen
- ✓ Hash-Chain-Verkettung — nachträgliche Manipulation ist erkennbar
- ✓ Filter nach Zeitraum, Aktion, Akteur, Objekt
- ✓ Export als CSV oder JSON
- ✓ ISO-27001- und SOC-2-konform aufgebaut
- ✓ Trennung von `audit_log` (sicherheitsrelevante Events) und `run_events` (Lifecycle-Events)
- ✓ Eigener Sidebar-Menüpunkt mit Listen-View und Verifikations-Skript

MODUL 10

Notifications

In-App und per Mail, granular konfigurierbar pro Event-Typ.

- ✓ In-App-Benachrichtigungen mit Top-Bar-Bell und 60-Sekunden-Polling
- ✓ E-Mail-Notifications für Zuweisungen, Eskalationen, Final-Approvals
- ✓ Sechs Trigger-Typen: `task_assigned`, `deadline_warning`, `run_started`, `run_completed`, `checkpoint_completed`, `escalation`
- ✓ Granulare Notification-Preferences pro User (in-app, mail, off) je Event-Typ
- ✓ Auto-Cleanup älterer Notifications durch den Background-Worker

MODUL 11

Sonstige Funktionen

Querschnittsfunktionen, die nicht eindeutig einem der zehn Hauptmodule zuordenbar sind.

- ✓ **Public-View** für Run-Detail per signiertem Token — externe Stakeholder können Status einsehen, ohne einen Account anzulegen (rate-limitiert auf 60 Requests/Minute pro IP, optionaler PIN-Schutz)
- ✓ **PDF-Export** für Run-Protokolle (drucktauglich, browser-basiert ohne externe Dependency)
- ✓ **Globale Suche** über Decks, Runs und User mit Deep-Links auf einzelne Objekte
- ✓ **Account-Settings** mit sechs Tabs: Profil, Passwort, Signatur-PIN, 2FA, Benachrichtigungen, Sicherheit
- ✓ **Avatar-Upload** mit Image-Resizing und Auth-only-Auslieferung
- ✓ **Backup-Service** mit automatischen SQLite-Snapshots durch den Background-Worker
- ✓ **Run-Schedules** für wiederkehrende Runs — drei Pattern-Typen: `daily`, `weekly`, `monthly`
- ✓ **SLA-Warnings** im Run-Detail mit Pre-SLA-Mahnung und Post-SLA-Eskalation
- ✓ **Datei-Archiv** mit Folder-Hierarchie, Datei-Sharing und Quota-System
- ✓ **Lead-Dashboard** für Department-Leads mit aggregierter Sicht auf eigene Abteilung

04 · ANWENDUNGSFELDER

Was sich abbilden lässt.

Die Plattform ist generisch konzipiert und nicht an eine bestimmte Branche gebunden. Jeder strukturierte Ablauf mit definierten Schritten, Verantwortlichkeiten und Fristen kann als Deck modelliert werden. Beispiele aus typischen Einsatzgebieten:

Mitarbeiter-Onboarding

Vertrag, Hardware-Bereitstellung, Account-Anlage, IT-Zugänge, Buddy-Zuweisung, erste Schulungen — als sequenzielle Lanes mit klaren Verantwortlichen und SLA-Fristen pro Phase.

Wartung & Inspektionen

Quartalsweise wiederkehrende Wartungsintervalle mit Run-Schedules, Prüfprotokollen, Datei-Anhängen (Fotos), erzwungenen Signaturen.

Kundenauftrags-Abwicklung

Aufmaß → Angebot → Auftragsbestätigung → Lieferung → Rechnung. Klare Übergaben, SLA-Stoppuhr pro Phase, Eskalation bei Überzug.

Compliance & Audits

Jährliche Hygiene-Schulungen, Brandschutz-Begehungen, DSGVO-Reviews, ISO-Audits — mit nachweisbarem Audit-Log und exportierbaren Protokollen.

Qualitätsmanagement

QM-Routinen, Stichproben, Ausschuss-Protokolle, Reklamationsabwicklung — reproduzierbar dokumentiert für ISO-9001-Audits.

Incident-Response

System-Ausfall, Datenleck, Lieferunterbrechung — ein Deck koordiniert sofort wer was wann macht, mit Eskalations-Logik und Sub-Prozessen für parallele Sub-Workflows.

Patientenpfade

Aufnahme, Anamnese, Voruntersuchung, Einwilligungen, Behandlung, Entlassung — strukturiert, signaturpflichtig, vollständig dokumentiert.

Bauabnahmen

Mehrstufige Abnahmen mit Sicherheitsprotokollen, Foto-Dokumentation, Mängelliste, Übergabe-Bestätigung — pro Gewerk eigene Lanes.

KOMPLEXITÄT WÄCHST MIT DEM BEDARF

Die **Lane-Engine mit parallelen Schritten**, die **Conditional Activation** und der **Sub-Prozess-Mechanismus** erlauben auch komplexe, verzweigte Abläufe — von einfachen Checklisten bis zu mehrstufigen, abhängigkeitsgesteuerten Workflows mit Approval-Gates.

05 · SICHERHEIT & COMPLIANCE

Sicherheit ist Fundament, nicht Feature.

Die Sicherheitsarchitektur von Prozessdeck ist mit einem Security-First-Ansatz entwickelt worden. Vier Audit-Pässe mit insgesamt mehr als 76 identifizierten und behobenen Findings haben das System auf produktive Belastbarkeit gebracht. Die folgenden Mechanismen sind im Code implementiert und verifizierbar.

AUDIT-PÄSSE	FINDINGS BEHOBEN	DAVON CRITICAL
4	76+	12

Jeder Request muss alle Schichten passieren — eine kompromittierte Schicht kompromittiert nicht das System.

Authentifizierung

- ✓ **Zwei-Faktor-Authentifizierung (2FA)** via TOTP nach RFC-6238, eigene Implementierung auf Standard-Library-Basis (HMAC-SHA1, Base32 ohne Padding für Authenticator-App-Kompatibilität)
- ✓ **Backup-Codes** für 2FA-Recovery — als Pbkdf2-Hash gespeichert, jeweils nur einmal verwendbar
- ✓ **Session-Management** mit URL-safen Tokens (32 Byte Entropie über `secrets.token_urlsafe`) — Sessions in dedizierter Tabelle, Logout-All durch Löschen aller User-Sessions

- ✓ **Pending-TOTP-Challenge-Tabelle** verhindert Brute-Force über mehrere Sessions hinweg
- ✓ **Auth-Audit-Logging:** jede Anmeldung, fehlgeschlagene Anmeldeversuche und Logout-Events werden im Audit-Log dokumentiert (mit Client-IP)
- ✓ **Signatur-PIN** pro User als zusätzlicher Schutz beim Datei-Signoff (8 – 32 Zeichen, Pbkdf2-Hash, Sperrung nach 5 Fehlversuchen)
- ✓ **Password-Reset-Flow** über zeitlich begrenzte Tokens (Tabelle `password_reset_tokens`) per E-Mail

Cookies & CSRF

- ✓ **Cookie-Hardening:** `Secure`-Flag (HTTPS only), `SameSite=Strict`, `HttpOnly`
- ✓ **CSRF-Schutz** per Session-Token: 32-Byte-URL-safe-Token in `sessions.csrf_token`, im `X-CSRF-Token`-Header bei allen Unsafe-Methods (POST/PUT/PATCH/DELETE) erforderlich
- ✓ **HMAC-Compare** für Token-Validierung — timing-safe gegen Timing-Attacks
- ✓ **CSRF-Whitelist** nur für Login, TOTP, Logout und Public-View — bewusst eng gefasst

Verschlüsselung

- ✓ **Fernet-Verschlüsselung** sensibler Felder (z. B. SMTP-Passwörter): AES-128-CBC + HMAC-SHA256 zur Authentizität
- ✓ **Master-Key** aus Environment-Variable oder auto-generiert in `app_settings.fernet_master_key`
- ✓ **Key-Rotation:** Roll-over zwischen alten und neuen Schlüsseln möglich, ohne Downtime
- ✓ **Format-Prefix** `"fernet:..."` erlaubt Round-Trip-Erkennung und Legacy-Klartext-Passthrough — Migrationen ohne Service-Unterbrechung
- ✓ **HTTPS/TLS** für die gesamte Verbindung zur Anwendung — Übertragung verschlüsselt, Schloss-Symbol im Browser

Rate-Limiting & DoS-Schutz

- ✓ **Sliding-Window Rate-Limiter** in-memory mit Thread-Lock und periodischem Garbage-Collection
- ✓ **Mehrere getrennte Buckets**: `login` (per IP+Username), `totp` (per User-ID — nicht per Pending-Token, sonst umgehbar), `account_auth`, `public_view`, `trigger_due` u. a.
- ✓ **TRUST_PROXY-Konfiguration** für korrektes `X-Forwarded-For`-Honoring hinter Reverse-Proxy — Spoofing-Schutz wenn nicht aktiv
- ✓ **Pre-Upload-Quota-Check** verhindert Disk-Exhaustion durch Power-User (Per-Tenant-Storage-Quota)
- ✓ **File-Size-Limit** hart auf 50 MB pro Datei
- ✓ **MIME-Type-Validierung** serverseitig — keine Verlässlichkeit auf Client-Header

Audit-Log

Jede sicherheitsrelevante Aktion wird in einem dedizierten Audit-Log persistiert: Authentication-Events, User-Mutationen, Deck- und Run-Lifecycle-Aktionen. Strikte Trennung zwischen `audit_log` (sicherheitsrelevante Events: Login-Versuche, User-Änderungen, Berechtigungen) und `run_events` (funktionale Lifecycle-Events: Run-Start, Status-Wechsel, Final-Approval).

- ✓ Lückenlose Erfassung aller sicherheitsrelevanten Aktionen mit Akteur, Objekt-Referenz und JSON-Details
- ✓ Filter nach Zeitraum, Aktion, Akteur, Objekt im Audit-View der Verwaltungs-Sektion
- ✓ IP-Adressen-Logging unter Beachtung von `X-Forwarded-For` bei aktiviertem `TRUST_PROXY`
- ✓ Indexierung über `created_at` und `(entity_type, entity_id)` für schnelle Abfragen
- ✓ Trennung zwischen `audit_log` (Sicherheitsereignisse) und `run_events` (Lifecycle-Events)

DSGVO-Konformität

- ✓ Alle Daten in einem **EU-Rechenzentrum** (Hostinger Frankfurt)
- ✓ **Auftragsverarbeitungsvertrag (AVV)** gemäß Art. 28 DSGVO mit Hosting-Anbieter
- ✓ AVV mit Endkunden auf Anforderung
- ✓ **Datenexport** jederzeit (CSV, JSON, PDF) — erfüllt Auskunfts- und Datenübertragbarkeitsrecht
- ✓ **Löschungsrecht:** nach Account-Kündigung 30 Tage Aufbewahrung, dann vollständige Löschung
- ✓ Datenschutzerklärung detailliert pro Modul
- ✓ Keine Tracking-Cookies, keine Analyse-Dienste, keine Drittland-Übermittlung

Architektonische Isolation

Im Multi-Tenant-Betrieb erhält jede Organisation eine **eigene Container-Instanz** mit dedizierter SQLite-Datenbank und eigenem Volume. Die Isolation ist **physisch, nicht logisch** — Cross-Tenant-Datenlecks sind durch fehlende gemeinsame Datenstrukturen architektonisch ausgeschlossen.

Was Tenants teilen

- ✓ Caddy-Reverse-Proxy mit TLS-Termination
- ✓ Host-Betriebssystem und Docker-Engine
- ✓ DNS-Subdomain-Routing

Was Tenants nicht teilen

- ✓ Datenbank — eigene SQLite-Datei pro Tenant
- ✓ File-Storage — eigenes gemountetes Volume
- ✓ Container — eigener Prozessraum, eigener Speicher
- ✓ Sessions, Auth-Tokens, Cookies

> **BELASTBARKEIT VERIFIZIERT**
>
> **4 Audit-Pässe mit insgesamt 76+ Findings** wurden seit MVP-Phase abgeschlossen. Davon **12 als CRITICAL klassifiziert** — z. B. Sub-Process-Cycle-Detection, Race-Condition bei parallelen Sub-Run-Inserts, TOTP-Bucket-Hijacking. Alle behoben und mit Tests gegen Regression abgesichert.

06 · TECHNOLOGIE & ARCHITEKTUR

Klein. Schnell. Wartbar.

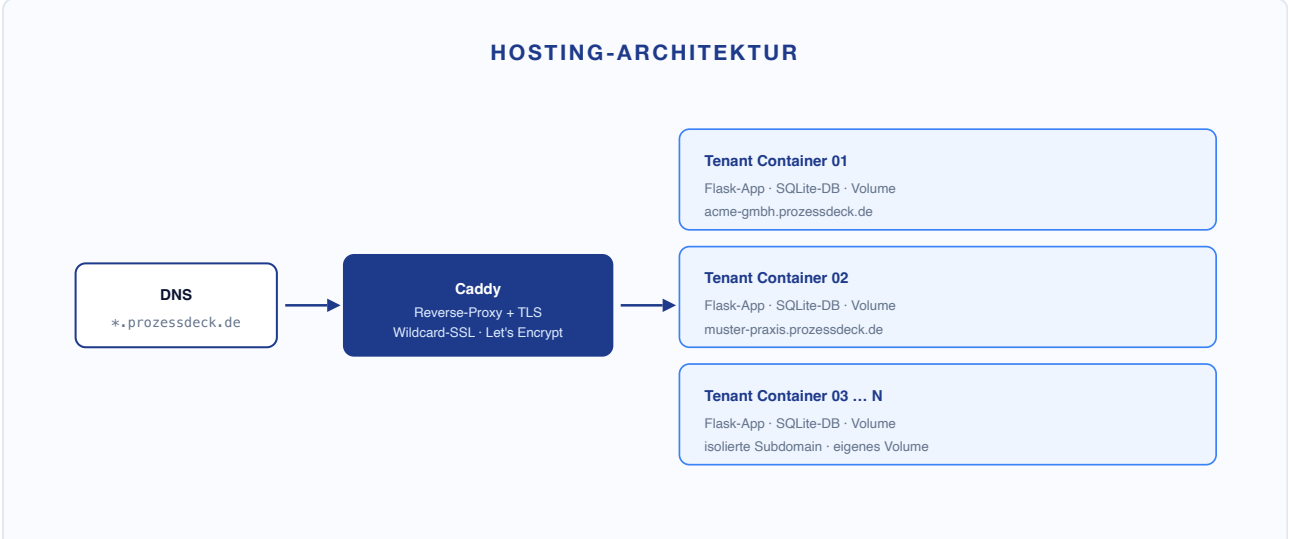
Prozessdeck setzt bewusst auf einen schlanken, langlebigen Tech-Stack ohne Framework-Hype und ohne Build-Pipelines. Das Ziel: in fünf Jahren noch wartbar zu sein, ohne ein einziges Major-Update einer JavaScript-Library zu überleben.

Tech-Stack im Überblick

Backend	Python 3.9+ mit Flask 3 (Blueprint-Architektur, 20 Routen-Module)
Datenbank	SQLite im WAL-Modus — Single-File, robust, atomar, kein DB-Server-Setup
Frontend	Vanilla JavaScript (ES-Module), eigene CSS-Tokens, kein Build-Step
Crypto	Eigene RFC-6238-TOTP-Implementierung, Fernet (cryptography-Lib), HMAC-SHA256
Mail	Python-`smtplib` (Standard-Library)
Background-Jobs	Daemon-Thread mit SLA-Engine, Reminders, Auto-Pruning, Backup
Build-Tools	Keine — kein Webpack, kein Bundler, kein Transpiler
Deployment	Docker-Container, Caddy als Reverse-Proxy, Wildcard-SSL via Let's Encrypt

Multi-Tenant-Architektur

Jede Organisation erhält eine eigene Container-Instanz mit dedizierter SQLite-Datenbank und eigenem Volume. Die Isolation ist physisch — Cross-Tenant-Datenlecks sind architektonisch ausgeschlossen.



Backend-Struktur

- ✓ **App-Factory-Pattern** in `app/__init__.py` mit Blueprint-Registrierung
- ✓ **20 Routen-Module:** account, admin, api, archive, audit, auth, calendar, checkpoints, decks, lead, notifications, overview, pdf, public_view, runs, schedules, search, stats, team, versions
- ✓ **21 Services:** activity, audit, backup, caldav_sync, calendar_invite, crypto, file_stamp, notifications_cleanup, notify, outlook_sync, pdf_export, pin, prozessdeck (Engine), quota, rate_limit, request_helpers, run_export_pdf, run_protocol, signatures, sla_worker, totp
- ✓ **Idempotente Migrationen** in `app/db.py` — Schema-Änderungen sind crash-safe
- ✓ **40 Tabellen** in vier Domänen: 18 Prozess-Kern, 8 Account / Auth / Rollen, 12 Sub-Module, 2 System

Frontend-Struktur

- ✓ **Single-Page-Application** mit Hash-Router (kein Server-Side-Rendering nötig)
- ✓ **Hauptseiten:** overview, decks, deck-editor, runs, run-detail, settings, calendar, stats, team, audit, archive, lead-dashboard
- ✓ **Wiederverwendbare Komponenten:** modal, user-select, share-dialog, run-start-dialog, schedule-dialog, checkpoint-dialog, version-compare-dialog
- ✓ **Mobile-Responsive:** Off-Canvas-Drawer-Sidebar, Agenda-View für Kalender, Touch-Targets ≥ 44 px, iOS-Auto-Zoom-Schutz
- ✓ **Eigenes Design-System** mit OKLCH-Farben, Inter + JetBrains Mono, kein Tailwind / Bootstrap

Background-Worker

Im Hintergrund läuft ein Daemon-Thread, der periodisch wiederkehrende Aufgaben übernimmt:

- ✓ **SLA-Engine:** Aktivierung von Schritten, Auswertung von Bedingungen, Stoppuhr-Berechnung — 5-Min-Tick
- ✓ **Reminder-Versand:** E-Mail-Erinnerungen vor SLA-Ablauf
- ✓ **Auto-Pruning:** alte Notifications, abgelaufene Sessions, abgelaufene TOTP-Challenges werden bereinigt
- ✓ **Backup-Routine:** automatische SQLite-Snapshots mit Versionserhalt
- ✓ **Run-Schedules:** Auslösung wiederkehrender Runs nach Zeitplan
- ✓ **3-Level-Eskalation:** Reminder → Assignee → Management bei SLA-Überzug

Wartbarkeit und Langlebigkeit

Bewusste Architektur-Entscheidungen für langfristige Stabilität:

- ✓ **Kein Build-Step im Frontend** — der Quelltext ist 1:1 das, was im Browser läuft. Debugging und Wartung ohne Source-Maps.
- ✓ **SQLite statt PostgreSQL/MySQL** — kein Datenbank-Server-Setup, atomare Backups durch Datei-Kopie, ausreichende Performance bis zu mehreren GB pro Tenant.

- ✓ **Standard-Library statt externer Bibliotheken** wo möglich — TOTP, SMTP, HMAC sind alles Python-Standard. Reduziert Dependency-Hell.
- ✓ **Keine Frontend-Frameworks** — Vanilla JavaScript altert nicht. Inter und JetBrains Mono sind Open-Source-Schriften ohne Vendor-Lock-In.
- ✓ **Idempotente Migrationen** — Schema-Änderungen können beliebig oft ausgeführt werden, ohne Zustand zu beschädigen.

> **PHILOSOPHIE**
>
> Jede Architektur-Entscheidung wurde nach dem Maßstab getroffen: *Was ist in fünf Jahren noch wartbar, wenn die heutigen Trends längst weitergezogen sind?* Die Antwort ist konsequent reduktiv — wenig Werkzeug, viel Substanz.

07 · KONTAKT

Für technische Rückfragen.

Bei weiterführenden Fragen zur Architektur, zu Sicherheits-Audits, DSGVO-Konformität oder Auftragsverarbeitung steht der folgende Kanal zur Verfügung.

Prozessdeck

EU-gehostet · Made in Germany · DSGVO-konform

E-MAIL

hallo@prozessdeck.de

WEB

prozessdeck.de

Was dieses Dokument beantwortet

- ✓ Vollständigen Funktionsumfang aller 11 Module
- ✓ Datenmodell und Engine-Logik
- ✓ Sicherheitsarchitektur und Audit-Stand
- ✓ DSGVO- und Compliance-Aspekte
- ✓ Technologie- und Hosting-Entscheidungen