Whitepaper · Technik, Sicherheit & Datenschutz

TherapieBegleiter – Sana KI-Assist Praxissoftware für Therapieberufe (Physio-, Ergo-, Logopädie, Podologie u. a.) · Stand: 28.05.2026

Dieses Dokument richtet sich an technisch und datenschutzrechtlich Verantwortliche (IT-Dienstleister, Datenschutzbeauftragte, Praxisleitung) und beschreibt Architektur, **Mandantenfähigkeit/Tenant-Isolation**, Sicherheitsmaßnahmen, Datenschutzkonzept, Betrieb sowie die geplante TI-/ePA-Anbindung ab 2027.

---

1. Architektur

Bereich	Umsetzung
Sprache/ Laufzeit	PHP 8.4 (Ziel 8.2+), bewusst **frameworklos**
Datenbank	MariaDB / MySQL, Zugriff ausschließlich über **PDO Prepared Statements**
Frontend	serverseitig gerendertes HTML, Vanilla CSS/JS – **keine** Frontend-Frameworks, keine Build-Pipeline
Webserver	Apache (vhost → `public/`), HTTPS
Struktur	PSR-4-Autoload `Praxis\` → `src/`, eigener Router, Controller/Model/View-Trennung

Designprinzip: Minimale Abhängigkeiten. Es gibt keine umfangreichen Drittbibliotheken im Anfrage-Pfad; selbst der E-Mail-Versand und der KI-Client sind abhängigkeitsfrei umgesetzt. Das verkleinert die Angriffsfläche und vereinfacht Wartung und Auditierung.

Schichten (vereinfacht):

```
public/index.php       → Front-Controller Praxis-App
                         (Tenant-Auflösung, Session-Start, Sicherheits-Header, Routing)
public-admin/index.php → Front-Controller Back-Office (Control-Plane)
                         eigener vhost (admin.<domain>), KEIN TenantContext
  Router/              → Pfad-Dispatch
  Middleware/          → AuthRequired, RoleRequired
  Tenant/              → TenantResolver, TenantContext, TenantRepository, Provisioner
  Admin/               → KundeRepository, KundeController, DemoController …
  Controller/          → Anwendungslogik je Modul (Tenant-scoped)
  Model/               → Repositories (PDO, Prepared Statements)
  View/                → Templates, Ausgabe-Escaping
```

Es gibt also **zwei getrennte Anwendungen** auf demselben Codestand:

1. die **Praxis-App** (`public/`) – läuft Tenant-isoliert, jede Praxis unter ihrer eigenen Subdomain mit eigener Datenbank;
2. die **Back-Office-App** (`public-admin/`) – mein internes Werkzeug zur Kundenverwaltung und Provisionierung der Demo-/Kunden-Instanzen, auf eigenem vhost (`admin.<domain>`) mit separatem Login und IP-Allowlist.

Details zur Mandantenfähigkeit folgen in § 2.

---

2. Mandantenfähigkeit & Tenant-Isolation

Mehrere Kundenpraxen laufen auf einer gemeinsamen Infrastruktur, aber **strikt voneinander getrennt**. Die Architektur-Entscheidung dafür ist **„Datenbank pro Tenant"** statt der häufiger gesehenen Variante „eine gemeinsame Tabelle + `tenant_id`-Spalte".

Warum DB-pro-Tenant (und nicht Row-Level):

Kriterium	**DB-pro-Tenant ✅**	**Row-Level (`tenant_id`) ❌**
Datenisolierung	hart auf DB-Ebene, **kein Leck durch vergessenes `WHERE`**	jedes fehlende `WHERE tenant_id` = Datenleck
Eingriff in Bestandscode	minimal: DB-Auflösung wird an einer Stelle (`Database::pdo()`) tenant-aware	jede Query / jedes Repository muss umgebaut werden
Backup je Tenant	eigener `mysqldump` je DB, eigene Verzeichnisse	`mysqldump --where` pro Tabelle, fehleranfällig
Teardown nach Demo	`DROP DATABASE` + `DROP USER`, fertig	tenant-weises Zeilen-Löschen, Risiko von Resten

Für Gesundheitsdaten ist die harte Isolation auf DB-Ebene das **risikoärmere Modell** – ein einzelner Bug kann nicht zum Cross-Tenant-Datenleck führen.

2.1 Auflösung des aktiven Tenants

Eine kleine **Control-Datenbank** (`praxis_control`) hält die **Tenant-Registry** (Tabelle `tenants`: Schlüssel, Subdomain, Praxisname, DB-Name, DB-User, Status, Ablauf, Kontakt-E-Mail).

Pro Request wird der aktive Tenant **vor** Session-Start und Routing aufgelöst:

- **Web:** aus `$_SERVER['HTTP_HOST']` → Subdomain → Registry-Eintrag.
- **CLI/Cron:** aus `--tenant=<key>` bzw. einer Tenant-Schleife (`--tenant=all`).

Findet sich keiner – oder ist der Tenant gesperrt/abgelaufen – wird eine eigene Fehlerseite ausgeliefert (`404` / `403`); **es gibt keinen Default-Tenant**, der versehentlich aktiv würde. Dieser „Crash-statt-Mischen"-Ansatz verhindert die häufigste Klasse von Multi-Tenant-Bugs.

2.2 Tenant-aware Datenbankzugriff

Die zentrale Methode `Database::pdo()` behält **ihre Signatur** – alle ~170 Aufrufe im bestehenden Code blieben unverändert. Was sich änderte: Sie liefert nun die Verbindung zur DB des **aktiven Tenants** aus `TenantContext::current()`, mit einem **Verbindungspool je Tenant-Schlüssel**. Cron-Schleifen über viele Tenants rufen `Database::forget($key)` zwischen zwei Tenants auf, damit nicht zu viele Verbindungen offen bleiben.

2.3 Dedizierter DB-User je Tenant (Least-Privilege)

Beim Anlegen einer neuen Praxis-Instanz erzeugt der Provisioner einen **eigenen MySQL-User** (`t_<key>_app`) mit einem **32-Hex-Zeichen Zufalls-Passwort** (128 Bit Entropie). Dieser User bekommt einen Least-Privilege-`GRANT` ausschließlich auf seine eigene Datenbank – **kein DROP, kein Cross-DB-Zugriff**. Damit hätte selbst ein kompromittierter App-Prozess einer Praxis keine Möglichkeit, andere Praxis-DBs zu lesen oder zu zerstören. Das eigentliche `DROP DATABASE` beim Demo-Teardown läuft über einen separaten, höher privilegierten **`control_user`**, dessen Zugang nur dem Provisioning-Werkzeug bekannt ist und nie in einem App-Request landet.

2.4 Sitzungs-Isolation je Tenant

Sitzungen sind bewusst **host-scoped**: Es wird **keine** `cookie_domain` gesetzt, das Cookie ist also nur unter der konkreten Subdomain gültig. Zusätzlich wird der `session_name` je Tenant abgeleitet. Dadurch ist eine versehentliche Cookie-Wiederverwendung über Tenant-Grenzen hinweg ausgeschlossen.

2.5 Provisionierung & Lifecycle

Eine neue Demo-Praxis entsteht über einen einzigen Aufruf (CLI oder Back-Office-Web-UI):

```
tenant:create <subdomain> <praxisname> [--days=14] [--ai-budget=10]
```

Schritte: Kapazitäts-Policy prüfen → `CREATE DATABASE` → dedizierter DB-User (siehe 2.3) → Schema migrieren (`migrate.php --tenant=<key>`) → Demo-Daten einspielen (`seed_testdata.php --tenant=<key>`) → Registry-Eintrag → Willkommens-Mail. Schlägt ein Schritt fehl, **rollt der Provisioner die Anlage vollständig zurück** (DB drop, User drop, Registry leer) – keine halbfertigen Tenants.

Ein täglicher **Lifecycle-Cron** kümmert sich um den Rest des Demo-Lebens:

Bedingung	Aktion
`aktiv` & `heute == ablauf − reminder_lead_days`	Erinnerungs-Mail „Demo endet bald"
`aktiv` & `heute >= ablauf`	Tenant auf `abgelaufen` (Login gesperrt, Daten bleiben)
`abgelaufen` & `heute >= ablauf + retention_days`	**finales Backup → `DROP DATABASE` / `DROP USER` → Registry-Eintrag entfernen**

Manuell gesperrte Tenants (`gesperrt`) bleiben unangetastet – die hält jemand bewusst.

2.6 Back-Office-CRM (Control-Plane-Web-UI)

Auf einem **eigenen vhost** (`admin.<domain>`) läuft eine kleine, **vom Mandanten-Produkt sauber getrennte** Back-Office-Anwendung (`public-admin/`, `src/Admin/`): Kundenliste mit Filtern, Kundenakte mit Pipeline-Status (Lead → Demo → Aktiv → Gekündigt), append-only Aktivitäts-Log und ein **Cockpit zur Provisionierung** (Demo anlegen / verlängern / sperren / entsperren / löschen) direkt aus der Kundenakte heraus. Login (Tabelle `admin_users` in der Control-DB) ist **getrennt** von den Tenant-Logins, der vhost ist per IP-Allowlist abgesichert, HSTS wird gesetzt.

---

3. Authentifizierung und Sitzungen

- **Passwörter** werden ausschließlich als **bcrypt-Hash** gespeichert (`password_hash`), niemals im Klartext.
- **Login** erzeugt eine frische Sitzung (`session_regenerate_id`) zur Vermeidung von Session-Fixation.
- **Sitzungs-Cookies:** `HttpOnly`, `SameSite=Lax`, `Secure` bei HTTPS.
- **Automatische Abmeldung** nach Inaktivität (Standard 30 Min) und eine harte Obergrenze (12 Std.); zusätzlich periodische Rotation der Sitzungs-ID. Bei Ablauf erfolgt ein Hinweis auf der Login-Seite.
- **Zugänge vergibt nur die Praxis** – keine Selbstregistrierung. Initialpasswörter werden generiert, per E-Mail versandt und einmalig angezeigt.
- **Passwort-Reset** als Self-Service über zeitlich begrenzte (60 Min), einmalig verwendbare Token; gespeichert wird nur deren SHA-256-Hash. Die Rückmeldung ist stets neutral (**Anti-Enumeration**: keine Auskunft, ob eine E-Mail existiert).
- **Login-Brute-Force-Bremse:** Eine eingebaute Drossel zählt fehlgeschlagene Logins im 15-Minuten-Fenster je IP-Adresse (Standard ≤ 20) und je Konto (≤ 10). Wer die Grenze überschreitet, wird mit neutraler Fehlermeldung blockiert; der Vorgang wird protokolliert. Die Zählung läuft **pro Praxis getrennt** und nutzt die Datenbank-Uhr (`NOW() - INTERVAL …`), damit Zeitzonen-Differenzen das Limit nicht aushebeln.

---

4. Autorisierung (Rollen- und Rechtemodell)

Vier Rollen – **Administrator, Inhaber, Therapeut, Patient** – steuern den Zugriff. Jede geschützte Route prüft serverseitig die erforderliche Rolle (`RoleRequired`). Beispiele:

- Patient:innen erreichen ausschließlich ihr eigenes Portal und **nur freigegebene** Dateien (sonst HTTP 403).
- Administrative Funktionen (Benutzer, Backups, Logs, DSGVO-Löschung) sind Administratoren vorbehalten; die Benutzerverwaltung schützt zusätzlich vor dem Aussperren des letzten Admins und vor Selbstsperre.

- Therapeut:innen sehen primär ihre eigenen Termine und Dokumentationen.

Die rollenabhängige Navigation ist nur die Anzeigeebene – maßgeblich ist stets die **serverseitige** Prüfung.

---

5. Schutz vor verbreiteten Angriffen

Risiko	Maßnahme
SQL-Injection	ausnahmslos PDO Prepared Statements
XSS (gespeichert/ reflektiert)	Ausgabe-Escaping in Views; Editor-Inhalte durch eine **strikte HTML-Allowlist** (`Html\Sanitizer`) bereinigt
CSRF	jedes Formular trägt ein Sitzungs-Token, jeder schreibende Endpunkt prüft es
Clickjacking / MIME-Sniffing	`X-Frame-Options`, `X-Content-Type-Options: nosniff`
Unsichere Datei-Uploads	Endungs-Allowlist **und** Inhaltsprüfung (finfo), Größenlimit 10 MB
Path Traversal	Downloads über realpath-Validierung gegen das Upload-Verzeichnis, erzwungener Anhang
Transport	HTTPS; HSTS-Header bei aktiver TLS-Verbindung

Sicherheits-Header werden zentral im Front-Controller gesetzt: Content-Security-Policy, X-Content-Type-Options, X-Frame-Options, Referrer-Policy, Permissions-Policy und (bei HTTPS) HSTS. Die CSP erlaubt derzeit bewusst `'unsafe-inline'` (Inline-Styles/-Handler im Kalender); eine spätere Härtung per Nonce ist vorgesehen.

Hochgeladene Dateien, Backups und Geheimnisse liegen außerhalb des Web-Roots. Konfigurationsdateien mit Zugangsdaten (Datenbank, E-Mail, KI-Schlüssel) sind von der Versionsverwaltung ausgeschlossen.

---

6. Revisionssicherheit der Dokumentation

Behandlungsdokumentation (SOAP) und Heilpläne folgen einem **Append-only-Prinzip**:

- Entwürfe sind frei bearbeitbar.
- **Abschließen** friert die Version unveränderlich ein (mit Angabe, wer wann).
- Korrekturen erfolgen ausschließlich als **Nachtrag** = neue Version; frühere Fassungen bleiben unverändert erhalten.

Auf abgeschlossene Versionen erfolgt im System kein UPDATE/DELETE. Damit ist der Behandlungsverlauf lückenlos nachvollziehbar.

Übersicht
Therapeuten
Patienten
Ärzte
Kalender
Abwesenheiten
Arbeitszeiten
Dokumentation
KI-Konfig
Meine E-Mails

Dokumentation

Zum Termin

Patient: **Jan Müller** Termin: 22.05.2026, 17:00 Uhr · Folgetermin Therapeut: Tobias Schäfer Status: Wahrgenommen

Abgeschlossen Version 1, abgeschlossen am 22.05.2026 17:00 Uhr

Subjektiv

Pat. klagt über morgendliche Steifigkeit, im Tagesverlauf besser. Pat. berichtet über Schmerzlinderung seit der letzten Behandlung.

Objektiv

ROM verbessert, Druckschmerz reduziert. Gangbild unauffällig. Schmerz aktuell VAS 3/10 (Vorwoche 5/10).

Analyse

Therapieansprache gut, Eigenübungen werden korrekt umgesetzt.

Plan

Kräftigung der Rumpfmuskulatur, Gangschule fortsetzen.

Heimprogramm:

- Dehnung ischiokrural 3×30 s
- Brücke 3×12
- LWS-Mobilisation in Rückenlage

Nachtrag verfassen

Dateien & Befunde

Noch keine Dateien hinterlegt.

Choose File No file chosen Hochladen

Erlaubt: PDF, DOCX, ODT, JPG, PNG, BMP · max. 10 MB.

Versionshistorie

Version 1 abgeschlossen

· tobias.schaefer@praxis-demo.de, 22.05.2026 17:00 Uhr · abgeschlossen von tobias.schaefer@praxis-demo.de

Subjektiv

Pat. klagt über morgendliche Steifigkeit, im Tagesverlauf besser. Pat. berichtet über Schmerzlinderung seit der letzten Behandlung.

Objektiv

ROM verbessert, Druckschmerz reduziert. Gangbild unauffällig. Schmerz aktuell VAS 3/10 (Vorwoche 5/10).

Analyse

Therapieansprache gut, Eigenübungen werden korrekt umgesetzt.

Plan

Kräftigung der Rumpfmuskulatur, Gangschule fortsetzen.

Heimprogramm:

- Dehnung ischiokrural 3×30 s
- Brücke 3×12
- LWS-Mobilisation in Rückenlage

7. KI-Addon: Architektur und Datenfluss

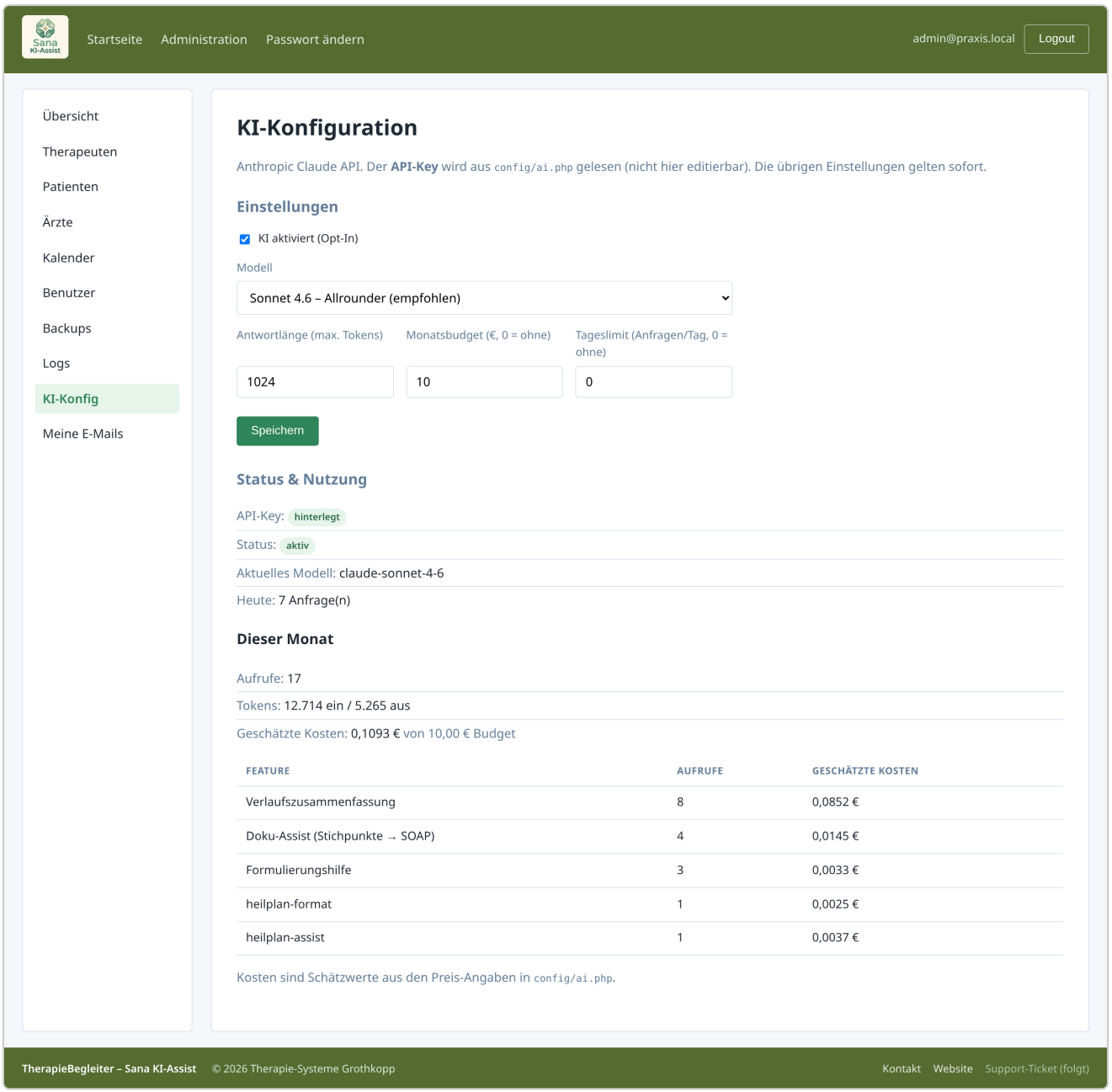
Das KI-Addon nutzt die **Anthropic Claude API** über einen schlanken, abhängigkeitsfreien HTTPS-Client (`src/AI/`). Es ist als **Opt-in** konzipiert und vollständig optional.

Komponenten:

- `Ai` – Konfigurations-Fassade. Lädt `config/ai.php`; fehlt die Datei oder ist kein API-Schlüssel hinterlegt, ist die KI **aus** und die Anwendung läuft normal weiter.
- `ClaudeClient` – einzelner Aufruf gegen `POST /v1/messages`.
- `Pseudonymizer` – entfernt Identifikatoren **vor** dem Versand.

Sicherheits- und Kostenkontrollen:

- Jeder KI-Endpunkt prüft Rolle, CSRF-Token und Aktivierungsstatus.
- **Pseudonymisierung vor dem Senden:** bekannte Eigennamen sowie E-Mail-Adressen und längere Ziffernfolgen (z. B. Versicherten-/Aktennummern) werden durch Platzhalter ersetzt; der klinische Inhalt bleibt erhalten. Das Verfahren ist **„best effort"** und ersetzt keine manuelle Sorgfalt.
- **Kein Auto-Speichern:** Die KI liefert nur einen Entwurf zur Anzeige; Übernahme, Bereinigung (Sanitizer) und Abschluss erfolgen über den normalen, von der Fachkraft gesteuerten Weg.
- **Budget und Limit:** weiche Monatsbudget-Grenze und hartes Tageslimit, in allen Endpunkten durchgesetzt; Verbrauch und geschätzte Kosten werden je Aufruf protokolliert.
- **Betreiber-Hard-Cap je Praxis:** Zusätzlich zum Praxis-Budget greift ein **vom Anbieter gesetzter Hard-Cap je Tenant** (`TenantContext::aiBudgetEur`). Er wirkt auch dann, wenn die Praxis ihr Eigenbudget auf „kein Limit" stellt – Schutz vor Kostenexplosion in der Demo- und Wachstumsphase. Der gedeckelte Wert ist in der KI-Konfigurationsansicht **read-only** sichtbar.
- **Schlüsselhaltung:** Der API-Schlüssel liegt ausschließlich in der gitignorierten Datei `config/ai.php` (Datei-Rechte `chmod 600`, Eigentümer `www-data`), **nie** in der Datenbank, **nie** im Git-Repository. Bei Verdacht wird er rotiert (Console des KI-Anbieters → alten Schlüssel widerrufen → neuen Schlüssel hinterlegen).



> **Datenschutzrechtlicher Hinweis (ehrlich eingeordnet):** Bei aktivierter KI werden die – pseudonymisierten – Inhalte an einen externen Dienstleister (Anthropic) übermittelt. Für den produktiven Einsatz mit Gesundheitsdaten ist daher ein **Auftragsverarbeitungs- vertrag (AVV/DPA)** mit dem KI-Anbieter sowie eine Bewertung des Drittlandtransfers erforderlich. Die Pseudonymisierung mindert das Risiko, ersetzt diese Prüfung aber nicht. Wer die KI nicht aktiviert, überträgt **keine** Daten an Dritte.

8. Datenschutz (DSGVO)

- **Datenminimierung:** für die KI nur pseudonymisierte Texte; keine Übertragung von Stammdaten an Dritte ohne Aktivierung.
- **Zweckbindung & Zugriffskontrolle:** rollenbasiert, serverseitig erzwungen.
- **Betroffenenrechte:** unwiderrufliche **Löschung (Art. 17)** inklusive Termine, Dokumentation, Heilpläne, Dateianhänge (Platte + Datenbank) und Login; protokolliert.
- **Patientenfreigabe:** Dokumente sind im Portal nur sichtbar, wenn die Praxis sie ausdrücklich freigibt.
- **Keine Drittanbieter im Frontend:** Schriftarten selbst gehostet (kein CDN/Google Fonts), keine Tracker.
- **Protokollierung:** sicherheits- und nachvollziehbarkeitsrelevante Ereignisse werden in einer Logtabelle erfasst.

---

9. Betrieb und Datensicherung

Hosting: Hetzner Cloud, Standort Deutschland (CCX23, dedizierte vCPU, 16 GB RAM, 160 GB NVMe), Debian 12, Apache + PHP-FPM + MariaDB. Wildcard-TLS-Zertifikat (Let's Encrypt DNS-01-Challenge) für `*.demo.<domain>`, Auto-Renewal. Cloud-Firewall: 80/443 öffentlich, 22 nur von der Admin-IP. Cloud-Backups als zusätzliches Sicherheitsnetz.

Backups je Tenant: `mysqldump` (gzip) in ein Verzeichnis außerhalb des Web-Roots, mit Rotation. Pro Tenant wird **ein eigenes Backup-Verzeichnis** geführt (`backups/<tenant-key>/`), damit die Rotation eines Tenants niemals die Sicherungen eines anderen löscht. Zugangsdaten werden über eine temporäre Optionsdatei statt über die Prozessliste übergeben. Bedienung per Cron **und** über die Admin-Oberfläche (erstellen, herunterladen, löschen) – jeweils tenant-gescoped. Zusätzlich wird die **Control-DB** (Tenant-Registry, Kunden-CRM) gesichert.

Restore-Drill: Vor dem Go-Live wird ein dokumentierter Wiederherstellungs-Lauf eines Tenants (Backup → frische DB → App startet sauber) als Abnahmekriterium durchgeführt – ein Backup, das nie zurückgespielt wurde, ist kein Backup.

Cron-Jobs (alle Tenants in einer Schleife):

Job	Frequenz	Inhalt
`backup.php --tenant=all`	nächtlich	Vollsicherung jeder aktiven Tenant-DB + Control-DB; Rotation gemäß `backup_keep`
`termin_erinnerungen.php --tenant=all`	morgens	Patienten-Erinnerungs-E-Mails für anstehende Termine
`tenant_lifecycle.php`	täglich	Demo-Lifecycle (siehe § 2.5): Erinnerung / Sperren / endgültig löschen

Das Schleifen-Muster (`set` → Arbeit → `Database::forget()`) verhindert, dass Verbindungen über viele Tenants akkumulieren; **ein scheiternder Tenant bricht die Schleife nicht ab** – die übrigen werden trotzdem versorgt, der Job endet aber mit Exit-Code ≠ 0, sobald irgendetwas gescheitert ist (für Monitoring auswertbar).

Transaktional-Mails an Tenants: Willkommens- und Erinnerungs-Mails laufen aus dem Provisioner bzw. dem Lifecycle-Cron über einen schlanken, abhängigkeitsfreien Mailer (Datei- oder SMTP-Transport). Jede gesendete Mail – ob automatisch oder aus dem Back-Office manuell ausgelöst – wird **im Aktivitäts-Log des verknüpften Kunden** protokolliert (Empfänger, Betreff, Status, Autor `system` oder Admin-E-Mail). Damit ist nachvollziehbar, ob/wann eine Praxis welche Mail bekommen hat – ohne separate Mail-Tabelle.

E-Mail-Transport: standardmäßig als `.eml`-Dateivorschau (kein Versand, DSGVO-freundlich für lokale Tests); produktiv per SMTP umstellbar. Für die Demo-Phase reicht ein Gmail-Relay (4 Slots, ~500 Mails/Tag); langfristig ist der Wechsel auf einen ESP (Postmark/Brevo) oder auf eigenes Postfix mit SPF/DKIM/DMARC eingeplant – beides ohne Änderungen am Anwendungscode.

Zeitzonen: Zeitkritische Vergleiche (24-h-Storno-Frist, Erinnerungsfenster, Login- Throttle-Fenster) werden bewusst in der **Datenbank** gerechnet (`NOW() - INTERVAL …`), um Abweichungen zwischen PHP- und DB-Zeitzone aus dem Weg zu gehen. Auf dem Server werden PHP- und MySQL-Zeitzone zusätzlich konsistent auf `Europe/Berlin` gesetzt.

---

10. TI / ePA / gematik (Pflichten ab 2027)

Die **Telematikinfrastruktur (TI)** und die **elektronische Patientenakte (ePA)** werden für Heilmittel-Praxen ab 2027 zur Pflicht:

Pflicht	Stichtag
TI-Anbindung Heilmittelerbringer	**1.10.2027** (Bundestag 6.11.2025)
eVO – elektronische Heilmittelverordnung	**1/2027**
ePA „für alle" (opt-out)	aktiv seit 2025 in der Fläche
TI 2.0 / HL7 FHIR verpflichtend	seit 2026

Strategische Entscheidung: NICHT selbst bauen. Die Anbindung ist **zertifizierungspflichtig** (gematik-Bestätigungs-/Zulassungsverfahren fürs Primärsystem) und benötigt **Hardware je Praxis** (eHBA, SMC-B, Kartenterminal). Für einen Solo-Anbieter ist die Eigenentwicklung weder wirtschaftlich noch zeitlich darstellbar. Die App wird stattdessen als **„Primärsystem"** gegen die dokumentierten FHIR-/Konnektor-Schnittstellen eines **zertifizierten TI-Gateway-/ePA-SDK-Anbieters** („TI as a Service") gebaut. Die TI 2.0 löst den physischen Konnektor durch ein **Cloud-Gateway** ab – das passt zur bestehenden Cloud-/Mandantenarchitektur.

Vorbereitung im Datenmodell (heute schon):

- **SOAP-Dokumentation** und **Heilpläne** liegen strukturiert vor (eigene Tabellen, Abschluss-Stati, append-only).
- **Verordnungen** enthalten **ICD-10-Code**, Diagnose-Klartext, Therapiebericht, Heilmittel und Restmenge – also genau die Felder, die später per FHIR übertragen werden.
- **Dateianhänge** an Patient/Termin/Heilplan sind als Content adressierbar (was später in/aus ePA fließen kann).
- **ICD-10-GM-Nachschlagehilfe** ist integriert (Mig.024, `icd10_gm` + `Icd10Repository`), Autocomplete am Diagnosefeld, amtlicher Code/Klartext.

Phasenplan (aktiv ab 2027):

Phase	Inhalt	Zeitfenster
P0	Positionieren / dokumentieren / Businessplan	läuft
P1	TI-Gateway-Anbieter sondieren, FHIR-Mapping spezifizieren, Zulassungs-/Kostenrahmen	Q4 2026 – Q1 2027
P2	ePA lesen/schreiben (Therapiebericht raus, Verordnung/Befund rein) + eHBA/SMC-B-Login je Praxis	Q2–Q3 2027
P3	eVO empfangen (digitale Verordnung statt Abtippen)	bis 10/2027
P4	KIM / TIM-Messaging	ab 2028

Bis dahin läuft die Demo-/Pilotphase 2026 bewusst **ohne TI** – das ist die Realität der meisten kleinen Praxen, die heute Verordnungen auf Papier bekommen und über externe Abrechnungszentren abrechnen. Wirtschaftliche Einordnung und Make-or-Partner-Begründung stehen im Businessplan (§ 8.1).

---

11. Qualitätssicherung

- Syntaxprüfung aller PHP- und JS-Dateien (`php -l`, `node --check`) als Routine.
- Repository-Abfragen werden gegen Demo-Daten verifiziert.
- Funktionsänderungen werden über HTTP-Durchläufe (echte Anfragen gegen die laufende Anwendung) end-to-end geprüft – inklusive der KI-Endpunkte.

---

12. Bekannte Einschränkungen / Ausblick

- **CSP-Härtung** per Nonce statt `'unsafe-inline'`.
- **SMTP-Versand** ist vorbereitet, Standard für lokale/Demo-Phase ist die `.eml`-Dateivorschau; produktiv via Gmail-Relay (Demo, ~500 Mails/Tag) bzw. ESP/Postfix (skalierend).

- **KI-Pseudonymisierung** ist „best effort"; für den Produktivbetrieb mit Gesundheitsdaten sind AVV/DPA und Drittland-Bewertung mit dem KI-Anbieter erforderlich.
- **Skalierungspfad:** Solange < ~15 Tenants auf der CCX23-Maschine laufen, bleibt die Architektur unverändert. Ab dann steht ein In-Place-Resize auf die nächste Cloud-Stufe an; ab Phase 3 (∞ Tenants) Trennung von Web-Node(s) hinter Load Balancer und dediziertem DB- Server – die DB-pro-Tenant-Architektur bleibt; das Limit „sehr viele DBs/Verbindungen" wird dort neu bewertet.
- **TI/ePA** (siehe § 10) ist aktiv ab 2027 geplant; bis dahin Demo-Betrieb ohne TI.
- Zielgeräte sind PC und Tablet; eine Smartphone-Optimierung ist nicht vorgesehen.
