MEDIVARIS SOFTWARE

RPMC

Ressource & Process Management Console

Technische Infomappe 2026 · Version 2.0

Für IT-Administratoren & Implementierer

Architektur · Anbindungswege · Sicherheit · Betrieb · KI-Architektur

On-Premise	ASP.NET Core 8	PostgreSQL 16	7 Wege
BETRIEBSMODELL	BACKEND-PLATTFORM	DATENHALTUNG	ORBIS-ANBINDUNG

medivaris Software · Inhaber Karl-Heinz Roth · Hersteller und Lizenzgeber · Version 2.0 · Stand 22.07.2026

Inhalt

Diese technische Infomappe beschreibt Aufbau, Anbindungswege, Sicherheitsarchitektur und Betrieb von RPMC aus der Perspektive der IT-Administration. Sie ergänzt die Produkt-Infomappe um das, was „unter der Haube" passiert — ohne vertrauliche Implementierungs-Interna.

1 Architektur-Überblick 3

2 Systemanforderungen & Sizing 6

3 Anbindungswege im Detail 8

4 Sicherheitsarchitektur 12

5 Betrieb: Deployment, Sync & Monitoring 15

6 KI-Architektur 20

7 Technischer Roadmap-Ausblick 22

A Anhang: Editionen, Portübersicht, Glossar 23

Hinweis zur Abgrenzung

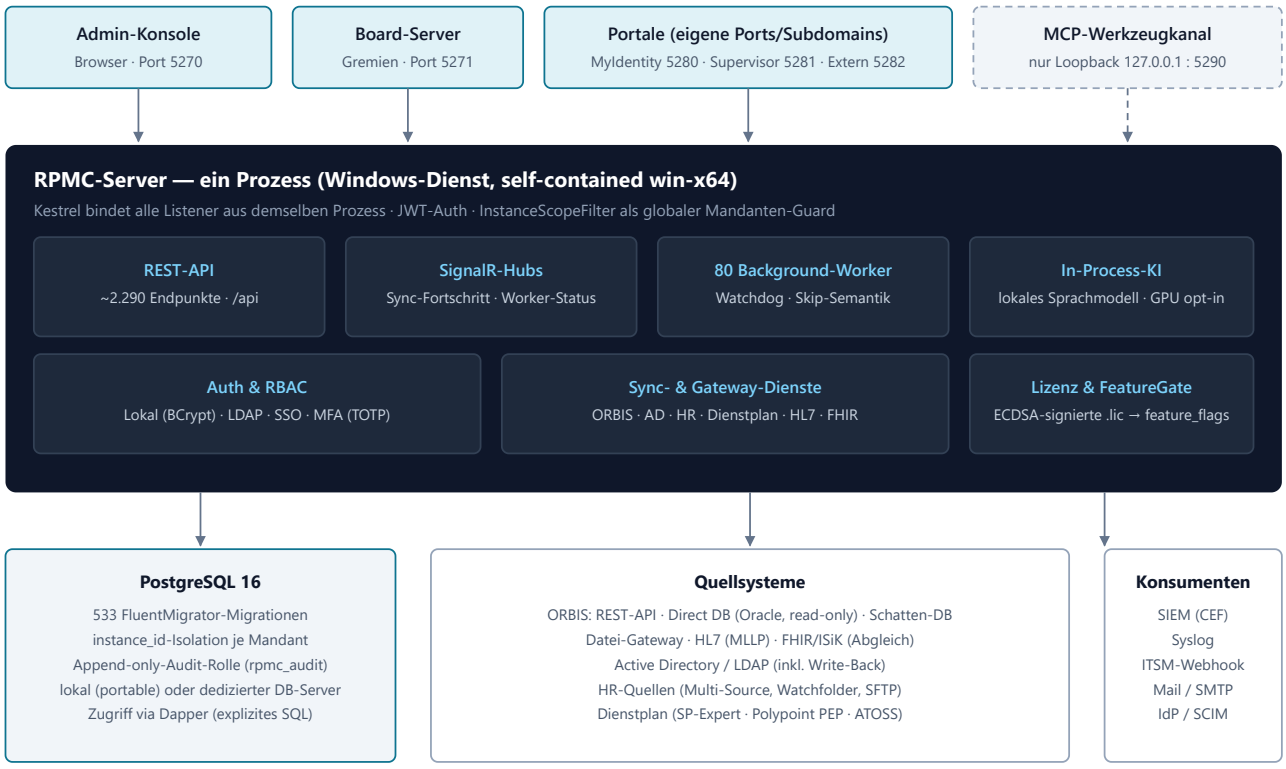
Produktstand 22. Juli 2026 (Version 2.0). Reifegrade: RELEASED = freigegebener Produktstand; BETA = eingeschränkter Reifegrad; IN ENTWICKLUNG = nicht Bestandteil des aktuellen Produkts; GEPLANT = Roadmap. Verbindlich sind ausschließlich Angebot und Vertrag.

KAPITEL 1

Architektur-Überblick

RPMC ist architektonisch ein **bewusst gewählter Monolith**: eine einzelne ASP.NET-Core-8-Anwendung, die REST-API, Frontend-Auslieferung, Echtzeit-Kommunikation, Hintergrund-Verarbeitung und Portal-Server in **einem Betriebssystem-Prozess** vereint. Für den Krankenhaus-Betrieb bedeutet das: ein Windows-Dienst, eine Datenbank, ein Update-Artefakt — keine Container-Orchestrierung, kein Message-Broker, keine Microservice-Landschaft, die eine Klinik-IT zusätzlich betreiben müsste.

ASP.NET Core 8 BACKEND · C# 12	**React 19** FRONTEND · TYPESCRIPT · VITE · TAILWIND 4	**PostgreSQL 16** DATENHALTUNG · DAPPER
533 DB-MIGRATIONEN (FLUENTMIGRATOR)	**80** BACKGROUND-WORKER + WATCHDOG	**bis 99** ORBIS-INSTANZEN JE INSTALLATION



> Abbildung 1 — RPMC-Systemarchitektur: ein Server-Prozess bedient Admin-Konsole, Board-Server und gebrandete Portale über getrennte Ports; PostgreSQL 16 ist der einzige Datenspeicher, Quellsysteme werden über dedizierte Gateway-Dienste angebunden.

1.1 Backend: ASP.NET Core 8 / C# 12

Das Backend besteht aus REST-Controllern (Namenskonvention `/api/v1/…`), Domain-Services und Repositories über **Dapper** als Datenzugriffsschicht. Die Entscheidung gegen einen Full-Feature-ORM (Entity Framework) und für Dapper ist bewusst: Jede Datenbank-Abfrage steht als explizites, reviewbares SQL im Code — für ein Audit-lastiges Produkt mit teils komplexen Governance-Abfragen ein Nachvollziehbarkeits- und Performance-Vorteil. Schreibzugriffe auf gemeinsam bearbeitete Entitäten nutzen **Optimistic Locking** (Zeilenversion + If-Match-Header, HTTP 409 bei Konflikt).

Der Server bindet mehrere Kestrel-Listener aus demselben Prozess: den Hauptport der Admin-Konsole, den Board-Server für Gremien sowie je einen eigenen Port pro Portal (MyIdentity, Supervisor, Externe Identitäten). Portale sind damit subdomain-fähig

hinter einem Reverse-Proxy und gleichzeitig über eine Port-Restriction-Middleware mit Pfad-Whitelist auf ihren Funktionsumfang beschränkt — ein Portal-Port kann konstruktionsbedingt keine Admin-API ausliefern. Kundenspezifische Portal-Ports werden aus der Datenbank (`portal_settings`) gelesen und beim Dienststart gebunden.

1.2 Frontend: React 19 / TypeScript / Vite / Tailwind 4

Das Frontend ist eine Single-Page-Application, die beim Build (`npm run build` via Vite) als statisches Bundle in das `wwwroot`-Verzeichnis des Servers gelegt und von dort ausgeliefert wird — es gibt keinen separaten Frontend-Server und kein Browser-Plugin. Styling erfolgt durchgängig über Tailwind-Utility-Klassen; die Oberfläche ist viersprachig (Deutsch, Englisch, Französisch, Italienisch; über 20.000 Übersetzungs-Keys je Sprache). Deutsch ist die vollständige Referenzsprache; einzelne Schlüssel in EN/FR/IT tragen noch einen Fallback auf Deutsch (Paritäts-Prüfung in Arbeit). Die aktive Mandanten-Instanz wird clientseitig in einem React-Context gehalten und bei jedem API-Aufruf als Header `X-RPMC-Instance-Id` mitgesendet.

1.3 Echtzeit: SignalR

Live-Zustände — Sync-Fortschritt, Worker-Status, Decision-Board-Bewegungen, proaktive KI-Insights, System-Health — werden über **SignalR** (WebSockets mit automatischen Fallbacks) an die Oberfläche gepusht. Administratoren sehen laufende Synchronisationen und Worker-Zustandswechsel ohne Browser-Refresh; das Frontend muss keine Polling-Schleifen fahren.

1.4 Worker-Modell: rund 80 Hintergrunddienste mit Watchdog

Periodische Aufgaben (Synchronisation, Risk-Scoring, Rezertifizierungs-Fristen, SoD-Prüfläufe, Mail-Versand, Aufräum-Jobs) laufen als **Hosted Services** im Server-Prozess — aktuell rund 80 Worker-Klassen (Registrierungen in `ServiceRegistration`). Ein zentraler **WorkerHealthService** überwacht sie:

- **Heartbeat-Watchdog:** Ein Watchdog-Timer prüft alle 5 Minuten jeden registrierten Worker (mit 5-Minuten-Karenz nach Serverstart). Der Health-Status eskaliert regelbasiert über `healthy → warning → degraded → critical` — abhängig von Fehlerzählern und davon, wie weit der letzte Heartbeat das Soll-Intervall überschreitet.
- **SIEM-Kopplung:** Worker-Ausfälle erzeugen SIEM-Events (`SYS-002` „Worker Failed", `SYS-003` „Worker Critical" mit hoher Severity) — die Betriebsmannschaft erfährt von einem Ausfall, bevor fachliche Daten veralten.
- **Skip-Semantik:** Ein Worker, dessen Modul per Feature-Flag deaktiviert ist, meldet explizit `skipped` statt eines falschen „success" — der Lauf gilt betrieblich als gesund, erhöht aber keine Erfolgs-Statistik.
- **Auto-Registrierung:** Beim Start durchsucht der Server den DI-Container und nimmt jeden Worker automatisch in die Überwachung auf — ein neuer Worker kann nicht „unsichtbar" bleiben.

1.5 Datenbank & Migrationen

Sämtliche RPMC-Daten liegen in einer PostgreSQL-16-Datenbank. Das Schema wird ausschließlich über **FluentMigrator-Migrationen** verwaltet — aktuell 533 versionierte Migrationen (V001 bis V533), die beim Serverstart automatisch und idempotent ausgeführt werden. Ein Update spielt also nie „lose SQL-Skripte" ein: Der neue Server-Build bringt seine Schema-Änderungen mit und zieht sie beim ersten Start nach (Details zum Update-Ablauf in Kapitel 5). Reflection-basierte Hygiene-Tests in der CI erzwingen lückenlose Versionsnummern und konsistente Migrations-Attribute.

1.6 Mandanten- und Instanz-Modell

Eine RPMC-Installation verwaltet mehrere **ORBIS-Instanzen** (Tabelle `orbis_instances`) — etwa die Häuser eines Verbunds oder Test-/Produktivsysteme. Die Isolation ist mehrstufig:

- **Datenebene:** Praktisch alle Datentabellen tragen eine `instance_id`-Spalte (in 297 der 533 Migrationen enthalten); jede Abfrage filtert darauf.
- **Zentrale Auflösung:** Der `InstanceContextService` löst die Ziel-Instanz je Request in fester Priorität auf — expliziter Parameter → Header `X-RPMC-Instance-Id` → Default-Instanz des Benutzers — und validiert den Zugriff gegen die Berechtigungstabelle. Fehlender Zugriff führt zu HTTP 403.

- **Globaler Guard:** Der `InstanceScopeFilter` läuft als globaler Action-Filter über *jeden* API-Aufruf und validiert Instanz-IDs aus Route und Query-String, bevor Controller-Code ausgeführt wird. Ausnahmen müssen explizit annotiert werden — Vergessen ist damit kein Sicherheitsrisiko mehr.

Superadmin-Konten (Berechtigung `*`) sehen instanzübergreifend; alle anderen Benutzer sind auf die ihnen zugewiesenen Instanzen beschränkt. Die Absicherung dieser Grenze durch Tests wird in Kapitel 4.3 beschrieben.

KAPITEL 2

Systemanforderungen & Sizing

RPMC ist für den **On-Premise-Betrieb auf einem Windows Server** im Krankenhaus-Netz ausgelegt. Die Auslieferung erfolgt als self-contained win-x64-Build — eine .NET-Installation auf dem Zielsystem ist nicht erforderlich. Der Betrieb erfolgt regulär als Windows-Dienst (`RPMC-Server`), alternativ portabel ohne Administratorrechte.

2.1 Hardware- und Software-Anforderungen

Komponente	Minimum	Empfohlen
Betriebssystem	Windows Server 2019	Windows Server 2022 (Linux-Betrieb möglich)
Laufzeitumgebung	keine — self-contained Build bringt .NET 8 LTS mit	
Datenbank	PostgreSQL 15	PostgreSQL 16 (mitgelieferte portable Instanz oder dedizierter DB-Server)
RAM	2 GB (Basisbetrieb ohne lokale KI)	4–8 GB; bei aktiver lokaler KI und großen Beständen entsprechend mehr
CPU	2 Kerne	4+ Kerne (parallele Sync-Worker, CPU-Inferenz der lokalen KI)
Festplatte	10 GB	50 GB inkl. Backups und Log-Retention
GPU (optional)	nur für beschleunigte lokale KI-Inferenz — siehe 2.4	

Richtwerte für den Basisbetrieb; das konkrete Sizing hängt von Mitarbeiter-Bestand, Anzahl Instanzen, aktivierten Direct-DB-Sub-Syncs und KI-Nutzung ab und wird im Implementierungsprojekt festgelegt.

2.2 Portable PostgreSQL mit automatischem Speicher-Tuning

RPMC kann die PostgreSQL-Datenbank **selbst mitbringen und verwalten** („Zero-Prerequisites-Deployment"): Der ServerManager initialisiert die Instanz (`initdb` / `pg_ctl`), registriert sie als eigenen Windows-Dienst und beschränkt sie auf localhost. Für problematische Windows-Umgebungen existiert eine mehrstufige Initialisierungs-Fallback-Kette (abgestufte `shared_buffers`-/Shared-Memory-Parameter), die auch auf gehärteten Servern zuverlässig durchläuft. Alternativ bindet RPMC einen extern verwalteten PostgreSQL-Server an.

Beim Serverstart wendet RPMC ein **RAM-bewusstes Speicher-Tuning** auf die Datenbank an (per `ALTER SYSTEM` + Konfigurations-Reload), damit große Bestände nicht auf initdb-Defaults laufen:

Parameter	Ableitung aus physischem RAM	Hinweis
`shared_buffers`	≈ 20 % des RAM	bewusst konservativ (Server teilt sich die Maschine mit dem .NET-Prozess und den Workern); greift nach PostgreSQL-Neustart
`effective_cache_size`	≈ 50 % des RAM	sofort wirksam
`maintenance_work_mem`	≈ 5 % des RAM	sofort wirksam
`work_mem`	RAM/512, gedeckelt	moderat wegen vieler paralleler Worker-Abfragen

Alle Werte sind nach oben und unten geklemmt (kleine Maschinen fallen nie unter die Defaults, große gefährden nicht den PostgreSQL-Start). Zusätzlich setzt RPMC Betriebs-Härtungen wie `wal_compression=on` (Schutz vor WAL-Wachstum bei ORBIS-Voll-Syncs) und `checkpoint_completion_target=0.9`. Für extern gemanagte, handgetunte Datenbanken ist das Auto-Tuning abschaltbar (`Database:AutoTunePostgres=false`).

2.3 Portkonzept

Ein einziger Server-Prozess bindet mehrere Listener; Portale erhalten eigene Ports und sind damit einzeln per Reverse-Proxy/Subdomain veröffentlichbar, ohne die Admin-Konsole zu exponieren:

Port	Dienst	Bemerkung
5270	Hauptserver (Admin-Konsole + REST-API)	eingehend; hinter Reverse-Proxy als HTTPS 443 veröffentlichen
5271	Board-Server (Decision Board)	separater Listener für Gremien-Zugriffe
5280	MyIdentity-Portal (Mitarbeiter-Self-Service)	je Portal eigener Port, gebrandete Login-Seite, Pfad-Whitelist (Port-Restriction-Middleware); kundenspezifische Zusatz-Ports aus `portal_settings` werden beim Dienststart gebunden
5281	Supervisor-Portal (Vorgesetzte)	
5282	Portal Externe Identitäten	
5290	MCP-Werkzeugkanal des RPMC-Assistenten	**ausschließlich Loopback** (127.0.0.1), nie öffentlich gebunden; nur bei gesetztem Freigabe-Token aktiv
5432 / 5433	PostgreSQL (Standard / portable Instanz)	nur intern (Server → DB); portable Instanz lauscht nur auf localhost

2.4 GPU-Option für die lokale KI

Die eingebettete KI (Kapitel 6) läuft standardmäßig **CPU-only** — für Assistenz-Funktionen mit kompakten lokalen Modellen ist keine GPU erforderlich (typische CPU-Inferenz: einige Token pro Sekunde; Antwortzeiten im zweistelligen Sekundenbereich). Für höheren Durchsatz oder größere Modelle ist GPU-Beschleunigung **pro Instanz opt-in** aktivierbar (CUDA-fähige Grafikkarte; konfigurierbare Anzahl der auf die GPU ausgelagerten Modell-Schichten über `gpu_layer_count`). Große Modellklassen setzen Karten mit ≥ 24 GB VRAM oder Server-Klasse-CPUs voraus. Die Modellauswahl bleibt herstellerneutral; Modelle werden als GGUF-Dateien mit dem Update-Paket ausgeliefert.

2.5 Deployment-Varianten

- **Windows-Dienst (Standard):** Installation nach `C:\RPMC`, Dienst `RPMC-Server`, portable PostgreSQL als zweiter Dienst; Verwaltung über das ServerManager-Werkzeug (Kapitel 5.1).
- **Portabler Modus:** Entpacken und starten ohne Administratorrechte — für Teststellungen und Evaluierungen.
- **Docker/Linux (alternativ):** Dockerfile und Reverse-Proxy-Konfiguration (Caddy) liegen bei; der primäre Support-Pfad ist der Windows-Dienst.

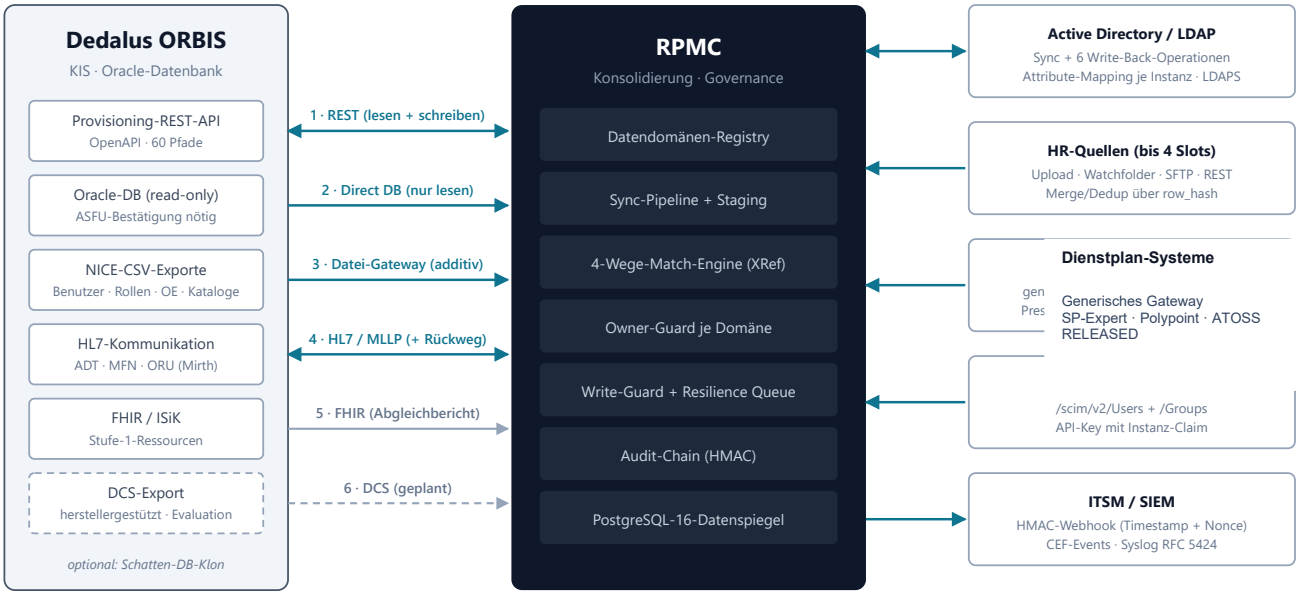
KAPITEL 3

Anbindungswege im Detail

RPMC konsolidiert Identitäts- und Betriebsdaten aus mehreren Quellsystemen. Für die ORBIS-Anbindung stehen mehrere, kombinierbare Datenwege zur Verfügung — welche Module welchen Weg benötigen, regelt eine zentrale **Datendomänen-Registry** (`DataDomainRegistry`): Jedes Modul deklariert seine benötigten Domänen; ist die zugehörige Quelle nicht aktiviert, wird das Modul in der Oberfläche ausgegraut bzw. ausgeblendet, statt leere oder falsche Daten zu zeigen.

Grundprinzipien aller ORBIS-Datenwege

- **Schreiben immer über die REST-API:** Schreibende Operationen Richtung ORBIS (Benutzer anlegen, Rollen zuweisen, Kontakte pflegen) laufen ausnahmslos über die offizielle Dedalus-Provisioning-REST-API — auch wenn Direct DB aktiv ist.
- **Direct DB ist strikt read-only:** Der Oracle-Direktzugriff erzwingt technisch SELECT/WITH-only; jeder Verstoß erzeugt ein SIEM-Event.
- **Owner-Prinzip:** Pro Datendomäne ist genau eine Quelle „Owner" des produktiven Datenspiegels. Zusätzliche Quellen (z. B. Datei-Gateway) ergänzen additiv, überschreiben aber nie den Owner-Bestand.



Doppelpfeile = bidirektional (REST-Schreibweg bzw. AD-Write-Back / HL7-Rückweg) · gestrichelt = nicht produktiver Owner

Abbildung 2 — Datenwege-Schaubild: sechs ORBIS-Kanäle (links) und die weiteren Quell-/Zielsysteme (rechts). Die Datendomänen-Registry entscheidet je Modul, welcher Kanal den produktiven Datenspiegel liefert.

3.1 ORBIS REST-API (Provisioning)

Der Standardweg. RPMC spricht die Dedalus-User-Provisioning-REST-API (OpenAPI 3.0.1, 60 Pfade / 76 Operationen). Ein interner Soll-Katalog (`OrbisApiEndpointCatalog`) mit Capability-Check prüft beim Verbindungstest, welche Endpunkte die konkrete ORBIS-Installation tatsächlich anbietet. Zugangsdaten werden verschlüsselt je Instanz gespeichert (`orbis_credentials`, eindeutig pro Instanz). Rund 60 % der RPMC-Module arbeiten vollständig REST-only; **alle schreibenden Operationen laufen ausschließlich über diesen Weg**.

3.2 ORBIS Direct DB (Oracle, read-only)

Für Datendomänen, die die REST-API nicht transportiert (Login-Audit, Fall-Ereignisse, Belegung, OE-Baum-Hierarchie, effektive Berechtigungen u. v. m.), liest RPMC direkt aus der ORBIS-Oracle-Datenbank — über `Oracle.ManagedDataAccess.Core`, mit technisch erzwungenem Read-only (nur SELECT/WITH; Verstöße erzeugen ein SIEM-Event). Die Verbindung wird je Instanz konfiguriert, der Connection-String verschlüsselt abgelegt. Über 20 einzeln aktivier- und planbare **Sub-Syncs** steuern feingranular, welche Domänen gelesen werden (Details in Kapitel 5.3).

> **Lizenz-Bestätigung des Betreibers (ASFU-Thematik)**
>
> Die ORBIS-Datenbank wird üblicherweise unter einer applikationsgebundenen Oracle-Lizenz (ASFU) betrieben. Ob ein lesender Direktzugriff durch Drittprodukte von der Lizenz des Hauses gedeckt ist, liegt in der Verantwortung des Betreibers. RPMC erzwingt deshalb eine explizite **Betreiber-Bestätigung**: Ohne dokumentierte Bestätigung (wer, wann, welche Textversion — persistiert in der Konfiguration) verweigert der Server die Aktivierung des Direct-DB-Zugriffs mit HTTP 422. Die Bestätigung wird in den Direct-DB-Einstellungen als Checkbox mit Erläuterungstext geführt.

3.3 Schattendatenbank / Offline-Klon

Für Test-, Schulungs- und Analyse-Szenarien ohne Dauerverbindung zur produktiven Oracle-DB existiert ein eigenständiges Werkzeug (**OrbisDumper**), das aus einem Massenexport einen strukturell originalgetreuen Klon der relevanten ORBIS-Schemata in einer lokalen Oracle-Instanz (Container) aufbaut — inklusive Tabellen, Daten, Fremdschlüsseln, Indizes, Sequenzen und Views, in referenzsicherer Phasenreihenfolge. Der Modus `--clone-full` erzeugt das Vollschema. RPMC kann gegen diesen Klon exakt wie gegen das Original arbeiten — für Implementierer der sichere Weg, Direct-DB-Auswertungen ohne Produktionszugriff zu validieren.

3.4 Datei-Gateway (ORBIS-NICE-Exporte)

Häuser, die keinen Direktzugriff aktivieren wollen, können offizielle ORBIS-NICE-CSV-Exporte einspeisen: Benutzer, Rollen inklusive Hierarchie, Rechte, OE-Struktur und Kataloge. Vier Transportwege stehen bereit: manueller Upload, überwachter **Watchfolder** (30-Sekunden-Poll), **SFTP-Pull** und REST-Pull; Duplikate werden über SHA-256-Hashes erkannt. Es gilt der **Owner-Guard**: Solange Direct DB für eine Domäne aktiv ist, ergänzt der Datei-Import den Datenspiegel nur additiv und überschreibt nie.

3.5 HL7 (MLLP-Listener)

Ein integrierter HL7-v2-Empfänger nimmt Nachrichten direkt aus der Kommunikationsdrehscheibe des Hauses an (z. B. Mirth): je Instanz ein TCP-Listener mit MLLP-Framing, 8-MB-Nachrichtenlimit, IP- und Absender-Allowlist (MSH-3), optionalem TLS und AA/AE-Acknowledgements. Verarbeitet werden **ADT A01–A50** (Aufnahme/Verlegung/Entlassung — speist die Raum-/Bettenbelegung near-realtime, ohne Historie), **MFN^M02** (Personalstammdaten), **MFN^M05** (Stationen/OEs) und **ORU**-

Für die in Entwicklung befindliche KI-Arztbrief-Pipeline ist HL7 als möglicher Rückweg vorgesehen

3.6 FHIR / ISiK

FHIR/ISiK ist in der aktuellen Version bewusst **kein produktiver Owner** einer Datendomäne: Der FHIR-Ingest liefert ausschließlich einen **Abgleichbericht** (Staging) — etwa um Practitioner-/Organization-Bestände gegen den RPMC-Spiegel zu prüfen. Hintergrund: Die ISiK-Basisprofile transportieren nur einen Bruchteil der Domänen, die RPMC benötigt. Der Ausbau folgt der ISiK-Reife der ORBIS-Installationen.

3.7 DCS (Data Collector Service)

Der von Dedalus als lizenzkonformer Datenexportweg vermarktete DCS wird als strategische Alternative zum Direct-DB-Zugriff geführt und je Kundensituation evaluiert. Er ist derzeit kein produktiver Datenweg in RPMC; Umfang und Konditionen liegen herstellerseitig.

3.8 Active Directory / LDAP

Der AD-Sync läuft in sechs Schritten (Connect, Fetch, Stage, Match, Merge, Snapshot) mit je Instanz konfigurierbarem **Attribute-Mapping** (Standard: `sAMAccountName`, `givenName`, `sn`, `displayName`, `mail`). Optional aktivierbar ist der **Write-Back** mit sechs auditierten Operationen: Konto deaktivieren/aktivieren, Passwort zurücksetzen (LDAPS-Zwang), Gruppe hinzufügen/entfernen, OU-Verschiebung — ergänzt um eine Konto-Anlage mit Dry-Run. LDAP dient zusätzlich als Login-Fallback inklusive Auto-Provisionierung von RPMC-Konten (Default-Rolle `viewer`). Seit Juli 2026 bündelt das **AD-Center** (eigene Navigationsgruppe) AD/LDAP-Anbindung, SSO, AD-Sync und AD-Administration.

3.9 HR-Quellen (Multi-Source)

Das HR-Gateway verwaltet **bis zu vier parallele HR-Quellen** (Slots) mit je eigenem Transport: Datei-Upload, Watchfolder, SFTP oder REST. Konnektor-Vorlagen für gängige HR-Systeme (u. a. Personio, SAP SuccessFactors/HCM, CSV-Sammelexport, generisches REST) setzen auf diesen Transporten mit vorbereiteten Feld-Mappings auf. Importe landen als Rohzeilen (JSONB) im Staging; die **Merge-Logik dedupliziert über Zeilen-Hashes** (`row_hash`) mit „newest-wins"-Konfliktstrategie, eine Quelle ist als führend (Master) markiert. Die 4-Tier-Match-Engine (exakt → normalisiert → fuzzy → Cross-Reference) verknüpft HR-Datensätze mit ORBIS-Identitäten.

3.10 Dienstplan-Systeme

RELEASED · Das generische Dienstplan-Gateway unterstützt REST-/SFTP-Transporte und Mapping in das RPMC-Datenmodell. Presets für SP-Expert, Polypoint PEP und ATOSS sind freigegeben; die konkrete Anbindung wird je Quellsystem konfiguriert.

3.11 ITSM / Webhooks

Ausgehende Ereignisse (z. B. Decision-Board-Statuswechsel) erreichen ITSM-Systeme als Webhook; eingehende ITSM-Aufträge nimmt ein dedizierter Endpunkt entgegen. Die Absicherung erfolgt dreistufig statt über JWT: **HMAC-SHA256-Signatur über den Raw-Body** (`X-Rpmc-Signature`), **Zeitstempel-Frischefenster ±5 Minuten** (`X-Rpmc-Timestamp`) und **Nonce-Replay-Schutz** (`X-Rpmc-Nonce`; Wiederverwendung wird mit 401 abgewiesen und im SIEM protokolliert).

3.12 SCIM 2.0

RPMC implementiert einen SCIM-2.0-Server (`/scim/v2/Users`, `/scim/v2/Groups`) für Push-Provisioning aus Entra ID, Okta oder OneLogin: eigenes Auth-Schema über API-Keys mit fest zugeordnetem Instanz-Claim (kein stiller Fallback auf eine Default-Instanz), SCIM-Filter-Parser, Paging und das Azure-übliche Deaktivierungsmuster (`active:false`).

3.13 Modul-Verfügbarkeit je Datenweg (Auszug)

Die folgende Matrix zeigt für ORBIS-datenabhängige Domänen, welcher aktivierte Weg sie produktiv versorgt (✓ = verfügbar, ○ = teilweise/eingeschränkt, — = über diesen Weg nicht verfügbar):

Datendomäne (Beispiele nutzender Module)	Nur REST	+ Direct DB	+ Datei-Gateway	+ HL7	+ FHIR
Mitarbeiter-Stammdaten, Konten, Rollen-Stamm, OE-Listen, Qualifikationen, Kataloge	✓	✓	○ (Rollen additiv)	—	—
Erweiterte Mitarbeiterfelder, Rollen-Hierarchie, effektive Berechtigungen, OE-Baum	—	✓	○ (Übernahme/Stichtag)	○ (OE geplant, MFN^M05)	—
Login-Audit (Compliance-Score, Sicherheits-Anomalien, Sitzungs-Ausschlüsse)	—	✓	—	—	—

Fall-Ereignisse (Fall-Zugriffs-Audit, Pfortenauskunft, Patientensuche-Überwachung, Access Insights)	—	✓	—	—	—
Belegung live (Raum-/Bettenbelegung)	—	✓ (inkl. Soll-Kapazität/Historie)	—	✓ (near-realtime, ohne Historie)	—
Belegungs-Historie (Kapazitäts-Reporting, OE-Nutzungs-Analyse)	—	✓	—	—	—
OA-Audit „Geändert durch" (Unified Audit), Verteilergruppen, OP-Management, PPR/TV-E, Konsile, Flow-Analytics, Kontakt-Tracing, Oracle-Live-SQL	—	✓	—	—	—
Arzt-Abgleich (LANR)	○ (Stamm + Link)	✓ (voll)	—	—	—

KI-Arztbrief-Quellen · IN ENTWICKLUNG · Direct DB: geplant · HL7: abhängig vom ORU-Feed

Kostenstellen	—	✓	✓ (additiv, Stamm)	—	—
FHIR/ISiK-Abgleichbericht	—	—	—	—	✓ (nur Staging)

Domänen ohne Häkchen in den rechten Spalten sind bewusst nicht über diese Wege verfügbar — kein Standard (REST, HL7, FHIR) transportiert diese Daten. Module, deren benötigte Domäne im gewählten Verbindungsmodus fehlt, werden in der Oberfläche ausgegraut. Vollständige, versionierte Matrix: `docs/DATENWEGE-MATRIX-MODULE.md`.

KAPITEL 4

Sicherheitsarchitektur

RPMC verarbeitet Berechtigungs- und Audit-Daten eines KRITIS-Betreibers — die Plattform selbst ist entsprechend gehärtet. Dieses Kapitel beschreibt Authentifizierung, Berechtigungsmodell, Mandanten-Isolation, Audit-Architektur, Plattform-Härtung, das Lizenzmodell und die Datenschutz-by-Design-Mechanismen.

4.1 Authentifizierung

Verfahren	Technische Umsetzung
Lokal	BCrypt-Passwort-Hashes mit konfigurierbarem Aufwandsfaktor (Work-Factor 12 für Seed-Konten). Account-Lockout: 5 Fehlversuche → 15 Minuten Sperre, auditiert.
LDAP-Fallback	Login prüft zuerst lokal, dann per LDAP-Bind gegen das AD. Optionales Auto-Provisioning unbekannter AD-Benutzer (`auth_source='AD'`, Default-Rolle `viewer`); AD-Konten führen kein RPMC-Passwort.
SSO	SAML2 (AuthnRequest, signaturvalidierte Assertions, `WantAssertionsSigned`, Single Logout) und OIDC (Discovery, Authorization-Code-Flow). Client-Secrets werden AES-verschlüsselt gespeichert; Claim-Mapping konfigurierbar, Auto-Provisioning mit `auth_source='SSO'`.
MFA (TOTP)	RFC-6238-Einmalcodes (6-stellig, 30-s-Fenster ±2 Schritte), 8 Einmal-Recovery-Codes. TOTP-Seeds und Recovery-Codes liegen AES-256-CBC-verschlüsselt in der DB. Eigener MFA-Lockout (5 Fehlversuche → 15 Minuten). MFA-Pflicht ist pro Rolle erzwingbar.
Sessions / JWT	Access-Token (HMAC-SHA256-signiert) mit bewusst kurzer Laufzeit von 30 Minuten; Refresh-Tokens 30 Tage mit **Rotation** (Altes wird bei Nutzung widerrufen) und JTI-Blacklist; Refresh-Tokens liegen nur als SHA-256-Hash in der DB.

4.2 RBAC: Rollen, Profile, Direktrechte

Fünf Systemrollen (`superadmin`, `admin`, `operator`, `auditor`, `viewer`) dienen als technischer Anker; die effektiven Funktionsrechte entstehen aus zwei Quellen über einen hierarchischen **Rechte-Baum** (Kategorien mit rekursiver Vererbung, Schema `domain:read|write|admin` mit Abwärts-Implikation):

- **Profile:** Benutzer → Profile → Berechtigungsrollen → Einzelrechte (granted/denied). Profile standardisieren Funktions-Bündel („Stationsleitung", „Auditor Fachabteilung").
- **Direktrechte:** gezielte Einzel-Overrides mit **Begründungspflicht**; das Superadmin-Recht ist per Direktgrant technisch ausgeschlossen.

Die aufgelösten Rechte werden beim Login als Claims in das JWT geschrieben und serverseitig pro Endpunkt geprüft. Zusätzlich erzwingt ein globaler **FeatureGateFilter** fail-closed die Lizenzgrenze: Endpunkte nicht lizenzierter Module antworten mit 403 (`feature_not_licensed`) — unabhängig von den Benutzerrechten.

4.3 Instanz-Isolation mit Architektur-Guard-Tests

Ergänzend zur in Kapitel 1.6 beschriebenen Mandanten-Isolation (zeilenweise `instance_id`, zentrale Auflösung, globaler `InstanceScopeFilter`) wird die Mandantengrenze durch eine **Regressions-Testsuite** abgesichert: rund zehn dedizierte Isolations-Testklassen (u. a. Scope-Enforcement-, Multi-Instanz-Isolations-, Cross-Tenant- und Delegated-Scope-Tests) stellen bei jedem Build sicher, dass Zugriffe auf fremde Instanzen mit HTTP 403 abgewiesen werden — auch über Nebenwege wie MCP-Werkzeuge, Gateways oder Portal-Endpunkte. Sicherheitsbefunde aus internen Pentest-Runden (SEC-Serie) sind als solche Tests fixiert, damit sie nicht regressieren.

4.4 Audit-Architektur

Hash-versiegelte Audit-Kette

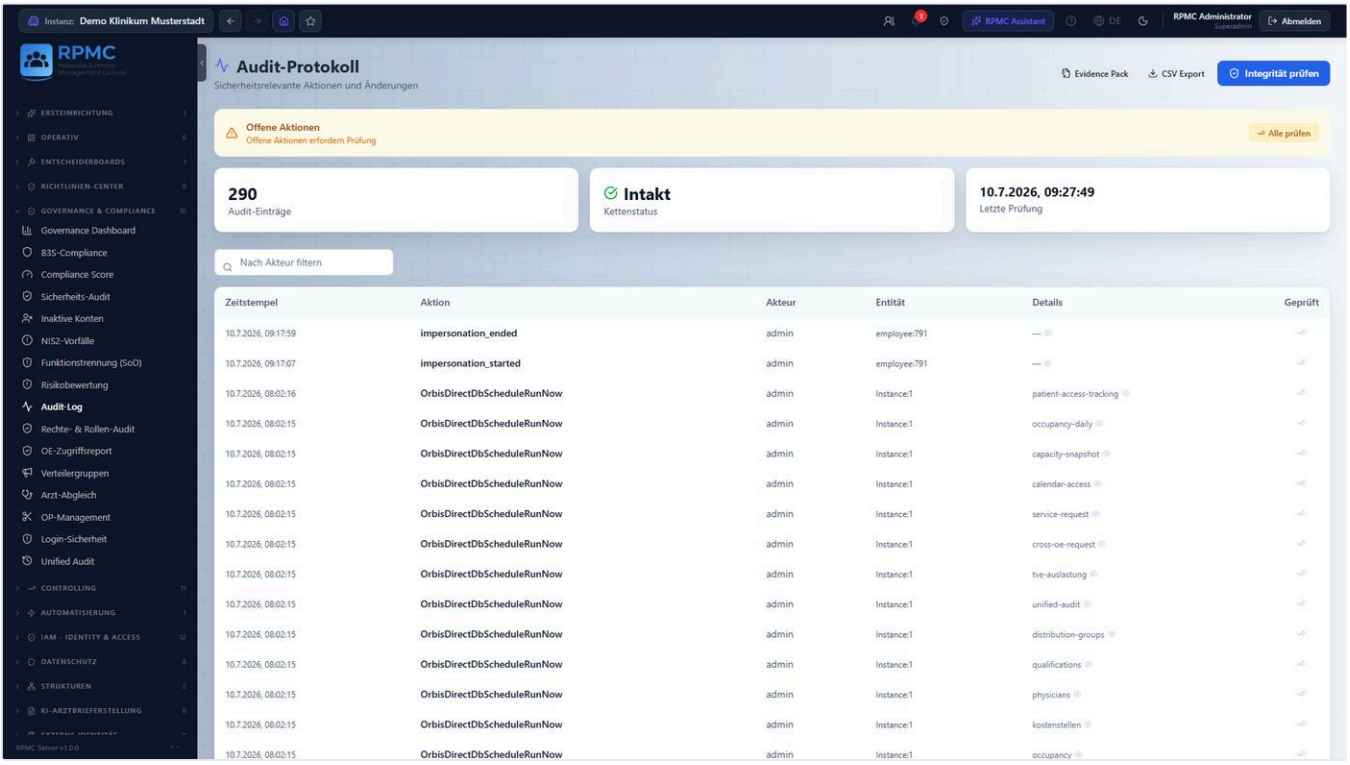
Sicherheitsrelevante Ereignisse werden in einer **tamper-evidenten Kette** gespeichert: Jeder Eintrag trägt `payload_hash` (SHA-256 des Inhalts), `previous_hash` (Verweis auf den Vorgänger) und `chain_hash` = HMAC-SHA256 über beide, mit fortlaufender Sequenznummer. Die Verifikation prüft die gesamte Kette; ein Bruch erzeugt sofort ein SIEM-Event höchster Severity (`AUDIT-001`). Geschrieben wird über einen dedizierten **Append-only-Datenbank-Benutzer** (`rpmc_audit`), dem UPDATE, DELETE und TRUNCATE auf den Audit-Tabellen auf DB-Ebene entzogen sind — Manipulation ist damit nicht nur verboten, sondern erkennbar.

Unified Audit

Die **Unified-Audit-Timeline** führt RPMC-Ereignisse und ORBIS-Ereignisse (aus dem Direct-DB-Audit-Sync) in einer chronologischen Ansicht zusammen — mit Quell-, Akteur- und Sensitivitäts-Kennzeichnung. Audit-Schreibfehler blockieren nie die Hauptoperation (non-blocking, aber protokolliert).

SIEM- und Syslog-Forwarding

Events verlassen RPMC wahlweise im **CEF-Format** (`CEF:0|medivaris|RPMC|…`) oder als **Syslog RFC 5424**. Weitergeleitet wird nach konfigurierbaren Filtern (Wildcard-Muster wie `AUTH-*`, `GOV-*`, `AUDIT-*`, `SYS-*`, `PROV-*`) und Mindest-Severity. Auch der Plattform-Eigenbetrieb meldet sich: Worker-Ausfälle erzeugen `SYS-002`/`SYS-003`-Events (Kapitel 1.4).



> Abbildung 3 — Audit-Protokoll mit Ketten-Status („Intakt"), Integritätsprüfung und chronologischer Aktionstabelle. Die Hash-Kette wird auf Knopfdruck über alle Einträge verifiziert.

4.5 Plattform-Härtung

- **Rate-Limiting:** Fixed-Window pro IP — global 200 Requests/Minute, Auth-Endpunkte (Login, Passwort-Reset, MFA-Verify) separat auf 15/Minute begrenzt; Überschreitung → HTTP 429.
- **Security-Header:** früh in der Pipeline (auch auf Fehlerantworten): HSTS, restriktive Content-Security-Policy (`default-src 'self'`, `frame-ancestors 'none'`), `X-Frame-Options: DENY`, `nosniff`, restriktive Permissions-Policy, Cross-Origin-Isolation.
- **CORS:** ausschließlich konfigurierte Origins; eine leere Liste in Produktion bedeutet Deny-all (kein `AllowAnyOrigin`).

- **Eingabe-Hygiene:** zentraler InputSanitizer (HTML-Stripping, Log-Sanitizing, Whitelist-Prüfung dynamischer SQL-Identifier); durchgängig parametrisierte Dapper-Queries.
- **PII-Disziplin in Logs:** E-Mail-Adressen werden in Logs maskiert bzw. gehasht; Klartext-Patientendaten gehören nie in Server-Logs.
- **Secrets:** JWT-Schlüssel, HMAC-Audit-Schlüssel und Verbindungs-Passwörter kommen aus Umgebungsvariablen bzw. dem lokalen Credential-Store; ORBIS-/AD-/SSO-Zugangsdaten liegen verschlüsselt (AES-256) in der DB.

4.6 Lizenzmodell

Lizenz-Dateien (`.lic`) sind **ECDSA-P-256/SHA-256-signiert** (kanonisches JSON, Unterstützung für Schlüssel-Rotation) und offline verifizierbar. Der Import erfordert Schema v2+; die Lizenz kann optional an Maschine bzw. Domäne gebunden werden (Maschinenname, Domain-SID, Volume-Serial, MAC — alles optional). Aus der Lizenz werden Edition, Limits (Mitarbeiter, Admin-Benutzer, Instanzen, Laufzeit) und die Modul-Whitelist gelesen und 1:1 in `feature_flags` geschrieben (License-as-SSoT, siehe Anhang A.1). Der Lizenz-Import ist ein Audit-Event; Overrides erfordern Begründung.

4.7 Datenschutz by Design

- **k-Anonymität in Analysen:** Aggregierte Zugriffs- und Cluster-Analysen weisen Kennzahlen erst ab einer Mindest-Gruppengröße aus (Konvention k = 5, z. B. mindestens 5 unterschiedliche Fälle pro Kandidaten-OE) — Einzelpersonen-Rückschlüsse aus Übersichten werden konstruktiv verhindert.
- **Pseudonymisierung:** Patient-Kontaktketten arbeiten mit Hash-Kennungen statt Klartext-Patienten-IDs; die Patientensuche-Überwachung (VIP-Schutz) liest nur datumsbegrenzte, aggregierte Spiegel. Vor jedem Cloud-KI-Aufruf ersetzt ein Pseudonymisierungs-Dienst Klarnamen durch Session-Token und übersetzt die Antwort zurück (Kapitel 6.2).
- **Stufen-Reveals mit Audit:** Besonders sensible Klartexte (z. B. die wörtliche Sucheingabe einer Patientensuche) erscheinen nie in Listenansichten. Erst ein expliziter, rollenbeschränkter Einzelabruf lädt genau einen Datensatz nach — und dieser Lesevorgang wird selbst auditiert (Audit-of-Audit). Auch jeder Lesezugriff auf Tracing-Gesundheitsdaten erzeugt ein SIEM-Event plus persistenten Audit-Eintrag.

KAPITEL 5

Betrieb: Deployment, Sync & Monitoring

5.1 ServerManager: Installation und Verwaltung vor Ort

Der **ServerManager** ist das Betreiber-Werkzeug für den gesamten Lebenszyklus der Installation: geführte Erstinstallation (PostgreSQL-Setup lokal oder extern, Windows-Dienst-Registrierung, Secret-Generierung, optionale Windows-/PostgreSQL-Härtung, HTTPS-Provisionierung mit Zertifikats-Diensten), Update-Import, Backups, Datenbank-Werkzeuge und Lizenz-Aktivierung. Der Installationszustand (installierte Version, Pfade, Ports, Betriebsmodus) wird in `rpmc_state.json` geführt.

5.2 RPKG-Patch-Deployment

Updates werden als **RPKG-Patch-Pakete** ausgeliefert — ZIP-Container mit Manifest (`patch_from`, `patch_to`, Datei-Aktionen add/replace/delete), Changelog und **Integritätssicherung**: SHA-256-Hash je Datei im Manifest, zusätzlich ein Root-Hash über alle Datei-Hashes sowie eine `.sha256`-Sidecar-Datei je Paket. Kryptographisch asymmetrisch signiert sind die Lizenz-Dateien; RPKG-Pakete sind hash-integritätsgesichert und versions-geschützt. Der ServerManager verifiziert das Paket beim Import fail-closed: jeder Archiv-Eintrag muss im Manifest deklariert sein, jede Datei-Prüfsumme (SHA-256) wird nachgerechnet, und Extraktions- wie Löschpfade werden gegen die Installationswurzel begrenzt (Schutz gegen Zip-Slip/Pfad-Ausbruch). Eine asymmetrische Herausgeber-Signatur des RPKG ist auf der Roadmap.

- **Sequenzieller Versions-Guard:** Ein Patch wird nur eingespielt, wenn `patch_from` exakt der installierten Version entspricht — Versionssprünge werden abgewiesen (dokumentierter Notfall-Override vorhanden; Backup und Rollback bleiben dabei aktiv). Fehlt eine Zwischenversion, wird ein kumulativer Patch oder ein Full-Paket gebaut; der Builder hält die letzten zwei Basis-Builds als Diff-Grundlage vor.
- **Zustandsschutz:** Konfigurations- und Zustandspfade (`appsettings*`, `rpmc_state.json`, `pgdata/`, `logs/`, Modelle, `.lic`/`.key`) werden bei Updates grundsätzlich nie überschrieben.
- **Backup & Rollback:** Vor jedem Einspielen: automatisches Pre-Update-Datenbank-Backup plus selektives Datei-Backup nur der berührten Dateien mit Restore-Manifest. Rollback stellt geänderte Dateien wieder her und entfernt neu hinzugekommene; die letzten drei Update-Backups werden vorgehalten.
- **Gesundheits-Gate:** Versionsstand und `installed_version` werden erst nach erfolgreichem Health-Check des neu gestarteten Dienstes gesetzt — ein „Version neu, Server tot"-Zustand ist ausgeschlossen. Vor dem Dienst-Stopp können laufende Synchronisationen kontrolliert auslaufen (Drain-/Wartungsmodus).

5.3 Datenbank-Migration im Betrieb

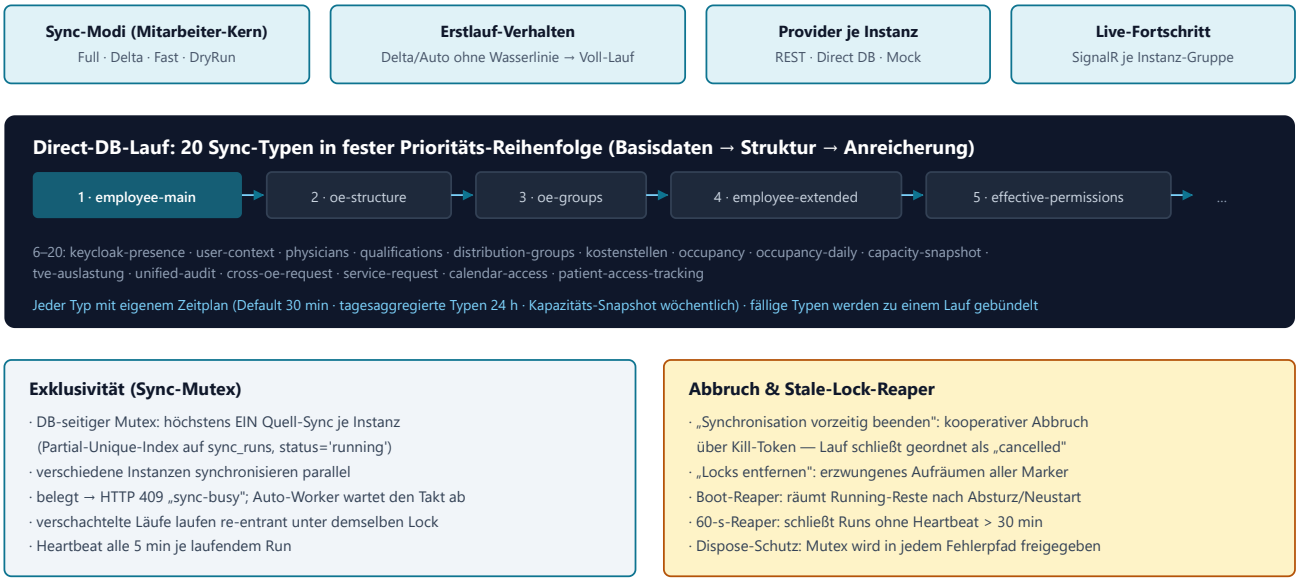
Schema-Änderungen reisen im Update-Paket mit: Beim ersten Start nach einem Update führt der FluentMigrator-Runner alle noch nicht angewendeten Migrationen automatisch aus (Admin-Verbindung), anschließend werden die Berechtigungen des Laufzeit-DB-Benutzers aufgefrischt (Dual-User-Strategie: Migrations-Admin vs. eingeschränkter App-User) und Seeds idempotent nachgezogen. Ein manueller SQL-Eingriff ist im Regelbetrieb nie erforderlich; für Rollbacks gilt der Weg über Backup-Restore (5.4).

5.4 Backups

- **In-App-Backups:** On-Demand über die API/Oberfläche — `pg_dump`-Export mit automatischem SQL-Fallback, SHA-256-Checksumme je Backup in der Historie, Verifikations-Endpunkt (erkennt Manipulation/Bitfäule), Download und Aufbewahrungsverwaltung.
- **ServerManager-Backups:** komprimierte `pg_dump`-Backups mit Zeitplan (täglich/wöchentlich, Uhrzeit, Aufbewahrung in Tagen), automatisches Pre-Update-Backup sowie ein **Voll-Klon**-Format (Haupt- und Audit-DB inklusive Manifest) für Umzüge und Testsysteme.
- Nach jedem Restore sollte die Audit-Ketten-Verifikation (Kapitel 4.4) ausgeführt werden.

5.5 Sync-Pipeline

Alle Quellsystem-Abgleiche laufen über eine zentrale, gegen Parallelität und Abstürze gehärtete Pipeline.



> Abbildung 4 — Sync-Pipeline: Modi und Erstlauf-Verhalten, Prioritäts-Reihenfolge der Direct-DB-Sync-Typen sowie die Schutzmechanismen Mutex, kooperativer Abbruch und mehrschichtiger Stale-Lock-Reaper.

Sync-Typen und Zeitpläne

Der Mitarbeiter-Kernabgleich kennt vier Modi: **Full** (Vollabgleich inkl. Zuordnungen und Löschungen), **Delta** (nur Änderungen seit der Wasserlinie; fällt beim Erstlauf selbstständig auf Full zurück), **Fast** (Bulk-Modus mit automatischem Rematch) und **DryRun** (reine Diff-Berechnung ohne Schreibzugriff — für gefahrlose Vorabprüfung). Im Automatikbetrieb wählt der Scheduler den Modus: Delta im Regelfall, Full bei Überschreitung des Reconcile-Intervalls oder fehlendem Erstlauf. Der Direct-DB-Pfad ergänzt **19 Sub-Syncs** mit je eigenem Zeitplan, DryRun-Schalter und Feature-Flag; ein Master-Schalter je Instanz und das globale Feature-Flag wirken als Kill-Switches. Große Oracle-Lesezugriffe laufen gebatcht (900er-ID-Blöcke, seitenweises Nachladen über Wasserlinien).

Locks, Abbruch und Selbstheilung

Ein datenbankseitiger Mutex stellt sicher, dass pro Instanz höchstens ein Quellsystem-Sync inklusive Matching läuft; parallele Startversuche erhalten HTTP 409. Neu ist die **kontrollierte Abbruch-Funktion**: Administratoren können einen laufenden Sync über die Verbindungsdiagnose vorzeitig beenden (kooperativer Abbruch — die Pipeline schließt den Lauf geordnet als „cancelled") oder mit „Locks entfernen" hängengebliebene Marker erzwungen aufräumen; beide Aktionen sind rechtegeschützt und auditiert. Dreifach abgesichert räumt der **Stale-Lock-Reaper** automatisch: beim Serverstart (Reste abgestürzter Läufe), minütlich (Runs ohne Heartbeat über 30 Minuten) und über einen Dispose-Schutz, der den Mutex in jedem Fehlerpfad freigibt.

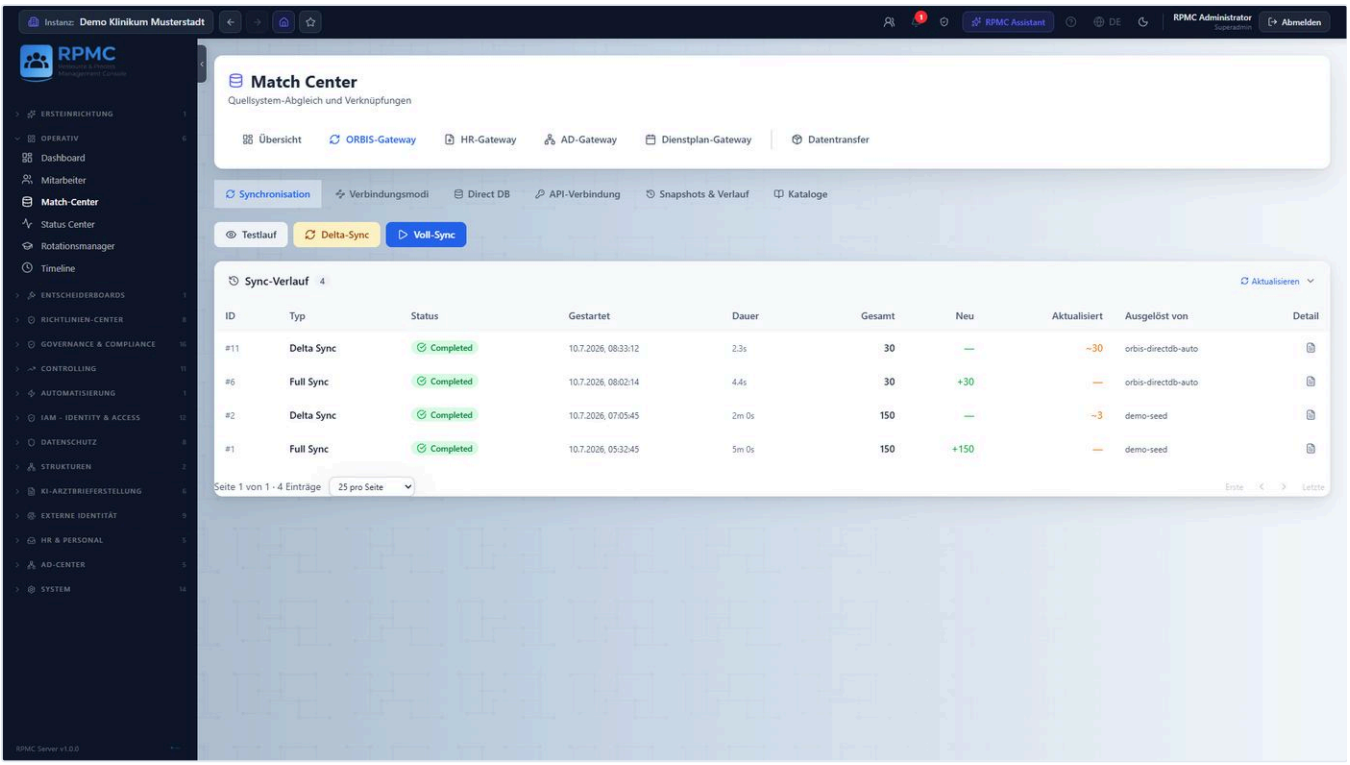


Abbildung 5 — ORBIS-Gateway: Sub-Tabs für Synchronisation, Verbindungsmodi, Direct DB und Kataloge; Testlauf-, Delta- und Voll-Sync sind manuell auslösbar, der Sync-Verlauf dokumentiert jeden Lauf.

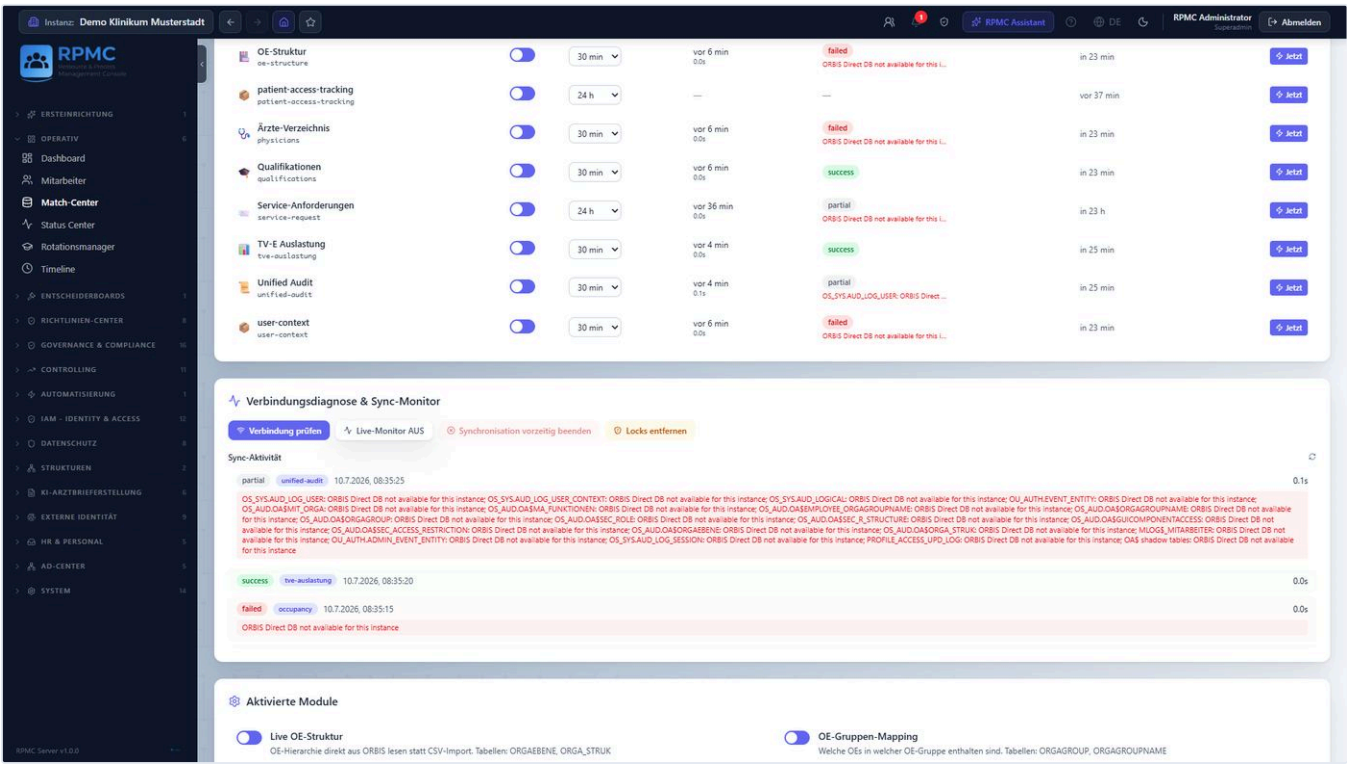
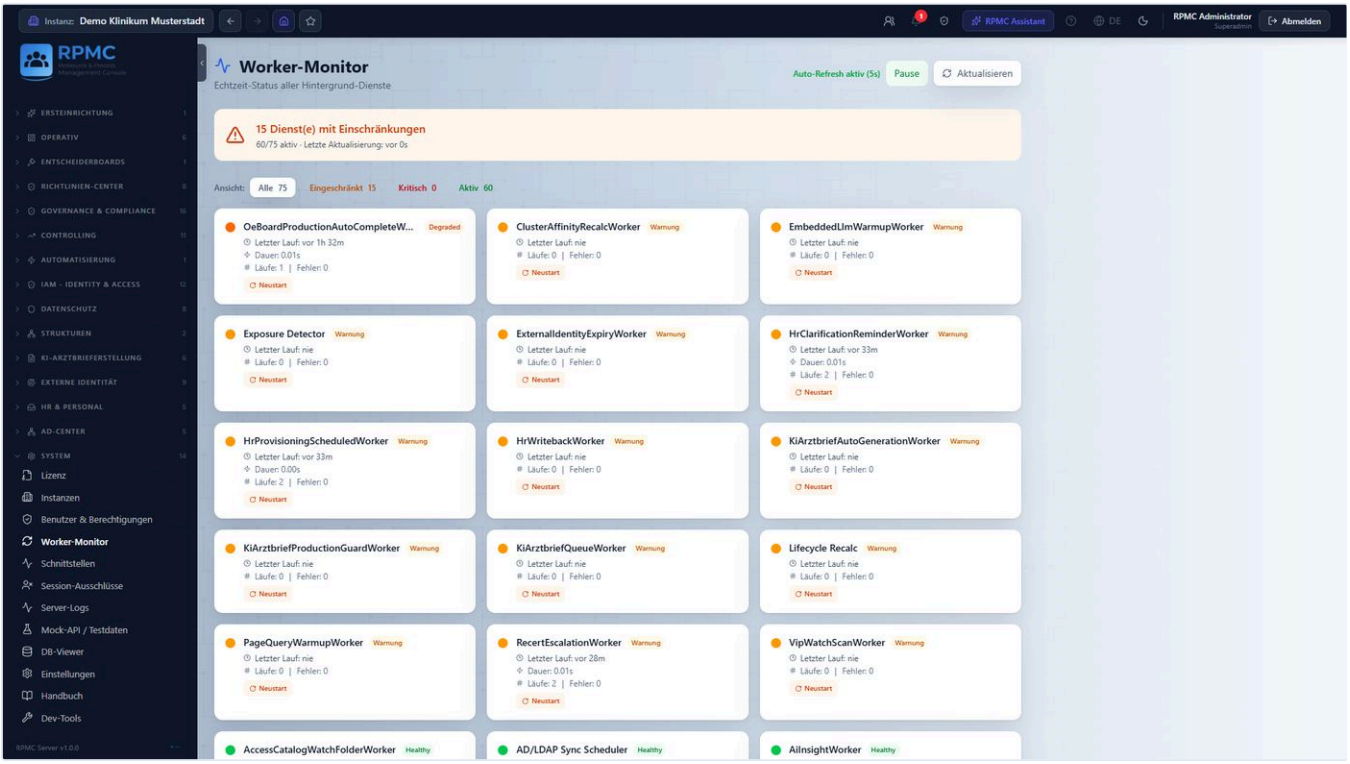


Abbildung 6 — Verbindungsdiagnose & Sync-Monitor mit Live-Sync-Aktivität und den Betriebsaktionen „Verbindung prüfen", „Synchronisation vorzeitig beenden" und „Locks entfernen".

5.6 Monitoring

- **Systemmonitor:** Ein Metrik-Sammler tastet CPU, RAM, Netzwerkdurchsatz und Datenträger alle 2 Sekunden in einen Ringpuffer (≈ 4 Minuten Historie) ab und speist das Live-Dashboard; ein Health-Aggregat fasst ORBIS (REST + Direct DB), AD, HR-Gateway, Mail, ITSM, Worker und Datenbank zu einem Gesamtstatus zusammen (Stale-Schwellen 24 h → „degraded", 48 h → „offline").

- **Worker-Monitor:** alle registrierten Hintergrunddienste mit Heartbeat, Fehlerzählern, letztem Exit-Grund und Diagnose-Hinweis; Zustandswechsel werden live per SignalR gepusht und als SIEM-Events gemeldet (Kapitel 1.4).
- **Logs:** In-Memory-Ringpuffer (1.000 Einträge) für die Log-Ansicht in der Oberfläche; rollierende Tages-Logdateien (`logs/rpmc-server-*.log`, 14 Tage, 100-MB-Grenze) plus separates Direct-DB-Log (30 Tage); optionaler Syslog-Forward für den KRITIS-Auditpfad. Health-Endpunkt (`/api/health`) für Nagios/Zabbix/Uptime-Monitoring.



> Abbildung 7 — Worker-Monitor im Systemmonitor: Status, Laufzeiten und Steuerung aller Background-Worker; „skipped" kennzeichnet per Feature-Flag deaktivierte Module.

5.7 Demo-Modus und Mock-ORBIS

Für Schulung, Teststellung und Entwicklung liefert RPMC zwei Betriebs-Hilfen mit:

- **Demo-Modus:** per Umgebungsvariable bzw. Konfiguration aktivierbar; seedet beim Start ein vollständiges Übungs-Szenario („Demo Klinikum Musterstadt": 750 Mitarbeiter über 14 Abteilungen, Rollen, HR-/AD-/Dienstplan-Daten, Audit-Trail, Demo-Benutzer). Ausgewählte produktionskritische Module sind im Demo-Modus zwangsweise deaktiviert.
- **Mock-ORBIS:** zwei Varianten — ein eigenständiger Mock-Server, der die ORBIS-Provisioning-REST-API simuliert (für Implementierungs-Tests ohne VPN/ORBIS-Zugang), und ein In-Process-Mock-Provider für den Direct-DB-Pfad, der als Mock markierte Instanzen bedient. Damit lassen sich Sync-Konfigurationen vollständig proben, bevor sie gegen die produktive ORBIS-Instanz freigegeben werden.

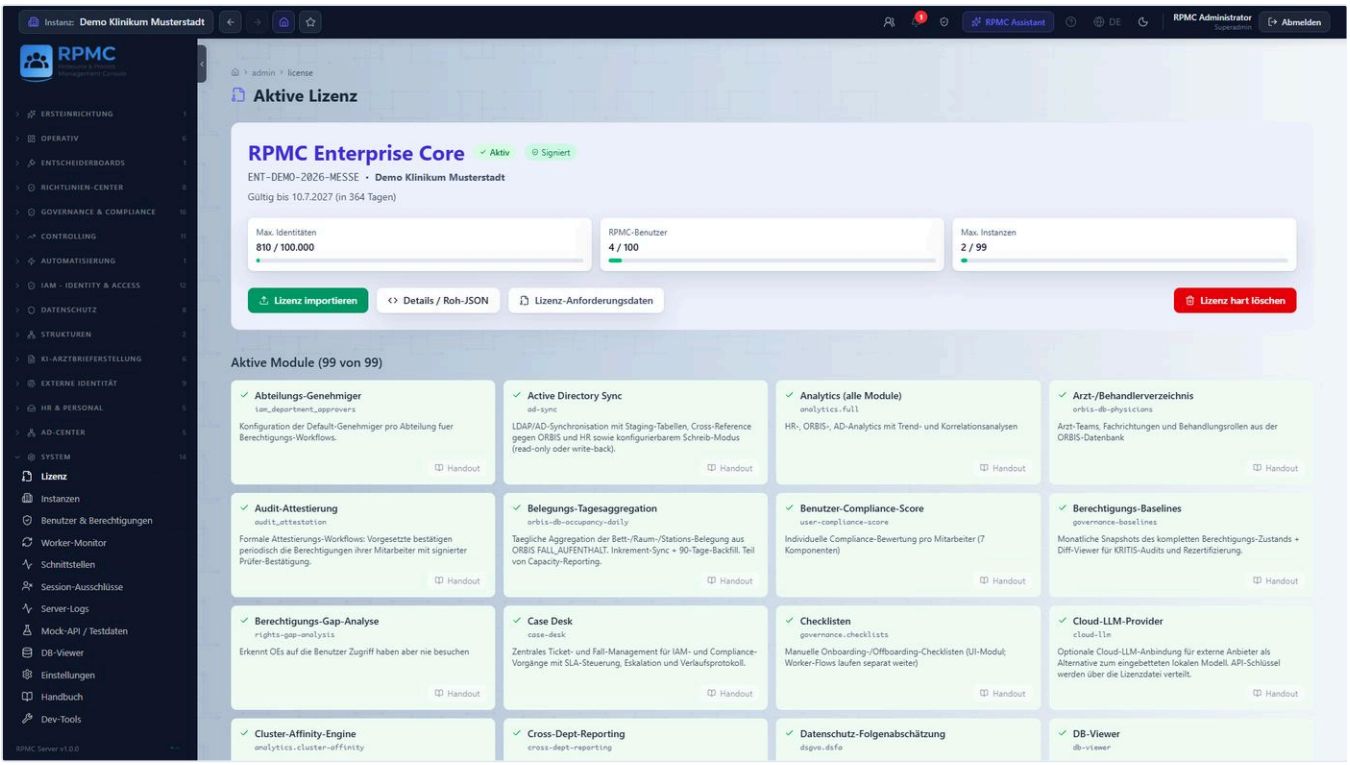


Abbildung 8 — Lizenzverwaltung: aktive Edition mit Limits (Identitäten, Benutzer, Instanzen), Gültigkeit und den aus der signierten Lizenz-Datei aktivierten Modulen.

KAPITEL 6

KI-Architektur

Die KI-Funktionen von RPMC sind konsequent auf Datensouveränität ausgelegt: Der Standardbetrieb kommt **ohne jede Cloud-Anbindung** aus, und wo Cloud optional zugeschaltet wird, stehen Pseudonymisierung und Auditierung technisch davor. Die Oberfläche ist herstellerneutral gehalten — es erscheinen keine Modell- oder Anbieternamen der lokalen Modelle.

6.1 In-Process-Inferenz (Standard)

Das lokale Sprachmodell läuft über eine **In-Process-Inferenzbibliothek im Server-Prozess selbst**: kein externer KI-Daemon, kein zusätzlicher Dienst, kein offener Port, keine Cloud-Pflicht. Modelle werden als GGUF-Dateien ausgeliefert (optional im RPKG-Update enthalten) und pro Instanz konfiguriert. Standard ist CPU-Inferenz; GPU-Beschleunigung ist pro Instanz opt-in (Kapitel 2.4). Inferenz-Aufrufe laufen strikt serialisiert über eine globale Semaphore — ein bewusster Single-Node-Kompromiss zugunsten von Speicher-Stabilität.

6.2 Optionale Cloud-Anbindung mit Schutzschicht

Für Häuser, die höhere Modellklassen nutzen wollen, existiert ein **Provider-Routing pro Instanz** (Default: eingebettet/lokal). Als dedizierter Enterprise-Cloud-Pfad ist **Google Vertex AI** angebunden — mit Service-Account-Authentifizierung (Schlüssel verschlüsselt, write-only) und technisch festgepinnter Region, sodass EU-Data-Residency (z. B. `europe-west4`) erzwingbar ist. Daneben lassen sich weitere Cloud- oder On-Premise-Inferenz-Endpunkte über ein kompatibles Standard-API-Schema anbinden, ohne neuen Code auszurollen. Vor jedem Cloud-Aufruf greifen drei Mechanismen:

- **Pseudonymisierung (fail-closed):** Klarnamen und Kennungen werden vor dem Versand durch Session-Token ersetzt (Token-Map mit Ablauf) und in der Antwort zurückübersetzt; ohne gültige Pseudonymisierungs-Session wird der Aufruf blockiert.
- **Outbound-Audit ohne Klartext:** Jeder ausgehende Prompt wird hash-basiert protokolliert (SHA-256 des Original- und des bereinigten Prompts, Zählwerte, optional pseudonymisierter Auszug) — es wird kein Klartext-Prompt persistiert; Aufbewahrung konfigurierbar.
- **Lizenz-Gate:** Cloud-KI ist ein eigenes Lizenzmodul (`cloud-llm`) und ohne explizite Freischaltung inaktiv.

6.3 RPMC-Assistent: Werkzeuge und Grenzen

Der eingebaute Assistent ist mehr als ein Chat: Er kann über eine kontrollierte Aktions-Schnittstelle in der Plattform arbeiten — navigieren, Cases anlegen, Rollen-/OE-Zuweisungen vorschlagen und ausführen, SoD-Regeln entwerfen, Compliance-Reports (B3S/NIS2/DSGVO/ISO 27001) erzeugen, Benachrichtigungen und E-Mails versenden sowie Workflows und Lifecycle-Regeln aus natürlicher Sprache generieren. Datenabfragen laufen ausschließlich **read-only**: Der Assistent darf nur SELECT-Abfragen ausführen, jede Abfrage wird zwangsweise auf die aktive Instanz gefiltert und in der Ergebnismenge begrenzt. Ergänzend liefern ein Wissensdienst (gelernte Muster, Peer-Analysen) und ein Insight-Worker (proaktive Hinweise per SignalR-Push) Kontext. Für Automatisierungs-Werkzeuge steht zusätzlich der MCP-Kanal bereit — ausschließlich auf Loopback gebunden und nur bei gesetztem Freigabe-Token aktiv (Kapitel 2.3).

6.4 KI-Arztbrief-Pipeline

IN ENTWICKLUNG

> **ABGRENZUNG**
>
> Die KI-Arztbrief-Pipeline ist eine eigenständige Entwicklung und kein Bestandteil des aktuellen RPMC-Produkts oder der derzeit angebotenen Editionen. Die nachfolgende Architektur beschreibt das Zielbild; Schnittstellen, Datenquellen und Lieferumfang können sich bis zur Freigabe ändern.

Geplante Pipeline-Bausteine

1. Kontrollierter Datenzug mit einzeln freizugebenden Quellen und PII-Einstufung.
2. Entwurfserzeugung über ein lokal oder gesondert freigegebenes Modell.
3. Ärztliche Prüfung und Freigabe in einem fallbezogenen Portal.
4. Konfigurierbarer Dokument-Rückweg über die Kommunikationsinfrastruktur des Hauses.

VOR PRODUKTIVER FREIGABE

Erforderlich sind technische Härtung, Datenschutzprüfung, klinische Validierung und eine gesonderte schriftliche Vereinbarung.

KAPITEL 7

Technischer Roadmap-Ausblick

Zwei größere technische Vorhaben sind aktuell in Entwicklung; sie werden hier bewusst als Ausblick und nicht als zugesagter Lieferumfang beschrieben. Vertragliche Termine werden projektspezifisch vereinbart.

7.1 Hochverfügbarkeit & Multi-Tenant für Klinikverbünde IN ENTWICKLUNG

RPMC ist heute ein bewusst zustandsbehafteter Single-Node-Dienst mit Mandanten-Isolation innerhalb einer Installation (bis 99 Instanzen). Für Klinikverbünde mit Verfügbarkeits-Anforderungen über einen Knoten hinaus ist ein Skalierungs-Zielbild in Arbeit, das auf einer dedizierten Architektur-Analyse basiert:

- **Stateless App-Knoten hinter Load-Balancer:** Auslagerung der In-Memory-Zustände (Record-Locks, Presence, Worker-Health) und geteilter Schlüsselring für die Datenschutz-APIs; verteilter Cache und Rate-Limiter.
- **SignalR-Backplane** für Live-Updates über mehrere Knoten.
- **Leader-Election für Worker:** Koordination der periodischen Worker über PostgreSQL-Advisory-Locks, damit Jobs clusterweit genau einmal laufen (der Sync-Mutex ist bereits heute datenbankseitig clustersicher).
- **PostgreSQL-Hochverfügbarkeit:** Replikations-Setup mit automatischem Failover und Connection-Pooling.

Die Multi-Tenant-Grundlagen (Instanz-Tabellen, Zugriffs-Grants, Lizenz-Limit `max_instances`, zentrale Instanz-Auflösung, Instanz-Claim im Token, zusammengesetzte Schlüssel `(instance_id, id)` auf Kerntabellen) sind bereits produktiv.

7.2 AD-Center IN ENTWICKLUNG

Das AD-Center bündelt die bisher verteilten Active-Directory-Funktionen in einer eigenen Navigationsgruppe: AD/LDAP-Anbindung und Verbindungstest, Single Sign-On (OIDC/SAML2), AD-Sync, AD-Administration (Benutzer-/Gruppen-Operationen aus RPMC heraus, inklusive Konto-Anlage mit Dry-Run) sowie ein integriertes Handbuch. Die Funktionsgruppe ist an das Lizenzmodul `ad-sync` gekoppelt und wird schrittweise ausgebaut — Ziel ist eine vollständige delegierbare AD-Operations-Konsole für den Krankenhaus-Betrieb.

7.3 Weitere technische Linien

- Dienstplan-Gateway für SP-Expert, Polypoint PEP und ATOSS: RELEASED; schichtbasierte Berechtigungsprozesse werden weiterentwickelt.
- KI-Arztbrief-Pipeline: separates Modul IN ENTWICKLUNG; nicht Bestandteil des aktuellen RPMC-Produkts oder der Editionen.
- FHIR/ISiK-Ausbau und DCS-Evaluation bleiben technische Entwicklungslinien.

ANHANG A

Editionen, Portübersicht, Glossar

A.1 Editions- und Modulmatrix

RPMC folgt dem **License-as-SSoT-Prinzip**: Die ECDSA-signierte Lizenz-Datei (`.lic`) ist die einzige Wahrheitsquelle für aktive Module. Beim Lizenz-Import iteriert der Server über den Modul-Catalog (Stand Juli 2026: 99 Module) und schreibt für jedes Modul die boolesche Entscheidung „lizenziert / nicht lizenziert" in die Tabelle `feature_flags`. Es gibt keine versteckten Editions-Defaults im Backend und keine Override-Logik im Code — eine Edition ist ein definiertes Modul-Set in der Lizenz-Datei.

Edition	Limits	Funktionsumfang (kumulativ)
Pilot Evaluierung	500 Mitarbeiter · 1 Instanz · 6 Monate	ORBIS-Anbindung (read-only), AD-Sync, HR-Import, Match Center (3-Wege), Status-Monitor Basis. Nicht enthalten: Decision Board, Workflows, Rezertifizierung, KI.
Core	1.500 Mitarbeiter · 1 Instanz · 15 Admin-User · 365 Tage	+ Status-Monitor erweitert (alle Kategorien), OE-Gruppen + OE-Struktur Import/Export, Mail-System, Schnellbereinigung mit Apply, Standard-Reports, RBAC mit Standard-Rollen.
Core Plus Hauptprodukt	5.000 Mitarbeiter · 3 Instanzen · 30 Admin-User · 365 Tage	+ Decision Board (Kanban/Voting/Quorum/Vier-Augen), Governance-Dashboard, Audit Enterprise mit HMAC-Integritätskette, ORBIS Rollen-Manager, Unified Audit, Compliance-Score, erweiterte RBAC-Rollen, Exportpakete für IT-Leitung/Datenschutz/Audit.
Enterprise Core Verbünde, KRITIS	100.000 Mitarbeiter · 99 Instanzen · 100 Admin-User · 365 Tage (Verbund: Sondervereinbarung)	+ Multi-Instanz (bis 99), Rezertifizierungs-Kampagnen mit Auto-Revoke, IAM Compliance Engine (NIS2/B3S/ISO 27001/DSGVO), Lifecycle-Engine (Joiner/Mover/Leaver), Delegated Admin, Audit Attestation, Overprivilege Detection, Capacity Reporting.
Trial	50 Mitarbeiter · 90 Tage	Evaluierungs-Lizenz gegen Mock-ORBIS; wird beim Erststart automatisch aktiviert.

> **EXTRA- und DISABLED-Overrides**
>
> Innerhalb einer Edition können einzelne Module per Lizenz-Override zusätzlich freigeschaltet (**EXTRA**, z. B. Cloud-KI in einer Core-Plus-Lizenz) oder gezielt deaktiviert werden (**DISABLED**, z. B. Audit Enterprise, wenn das Haus eine eigene Audit-Pipeline betreibt). Jeder Override wird auditiert; Overrides können ausschließlich Module aktivieren, die in der signierten Lizenz enthalten sind.

A.2 Portübersicht (Schnellreferenz)

Port	Dienst	Richtung	Exposition
5270	Hauptserver (Admin-Konsole, REST-API, SignalR)	eingehend	intern; extern nur via Reverse-Proxy (HTTPS 443)
5271	Board-Server (Decision Board)	eingehend	intern / je Bedarf via Proxy
5280	MyIdentity-Portal	eingehend	je Portal eigene Subdomain möglich; Pfad-Whitelist
5281	Supervisor-Portal	eingehend	wie 5280
5282	Portal Externe Identitäten	eingehend	wie 5280
kundenspez.	zusätzliche Portal-Ports aus `portal_settings`	eingehend	werden beim Dienststart gebunden
5290	MCP-Werkzeugkanal (RPMC-Assistent)	lokal	**nur Loopback 127.0.0.1**; nur mit Freigabe-Token aktiv

5432	PostgreSQL (Standard-/Extern-Installation)	intern	Server → DB
5433	PostgreSQL (portable, von RPMC verwaltet)	intern	nur localhost
514	Syslog-Forwarding (optional)	ausgehend	UDP, RFC 5424/3164 an SIEM-Kollektor
636/389	LDAPS/LDAP zum Domain-Controller	ausgehend	LDAPS-Zwang bei Passwort-Operationen
1521	Oracle (ORBIS Direct DB, read-only)	ausgehend	nur bei aktiviertem Direct-DB-Weg
konfigurierbar	HL7-MLLP-Listener (je Instanz)	eingehend	IP-/Absender-Allowlist, optional TLS

A.3 Glossar technischer Begriffe

Append-only — Datenbank-Modus, in dem nur Anfügen erlaubt ist; UPDATE/DELETE/TRUNCATE sind auf DB-Ebene gesperrt (Audit-Tabellen).

ASFU — Application Specific Full Use; Oracle-Lizenzform der ORBIS-Datenbank. Direktzugriffe außerhalb der Applikation berühren die Lizenzfrage des Betreibers.

Audit-Chain — HMAC-verkettete, manipulationserkennende Folge von Audit-Einträgen.

BCrypt — Passwort-Hash-Verfahren mit konfigurierbarem Aufwandsfaktor.

Break-the-Glass — Auditierter Notfall-Zugriffsmechanismus mit zeitlich begrenzter Rechteerhöhung.

CEF — Common Event Format; Standard-Ereignisformat für SIEM-Systeme.

Dapper — Schlanker .NET-O/R-Mapper mit expliziter SQL-Kontrolle (statt EF Core).

DAG — Directed Acyclic Graph; Topologie der Workflow-Engine.

Delta-Sync — Synchronisation nur der Änderungen seit dem letzten Lauf.

DirectDB — Lesender Direktzugriff auf die ORBIS-Oracle-Datenbank (statt REST-API).

DryRun — Simulationslauf ohne Schreibzugriff; zeigt Änderungen vorab an.

ECDSA P-256 — Signaturverfahren auf Basis elliptischer Kurven; signiert die Lizenz-Dateien. RPKG-Update-Pakete sind demgegenüber hash-integritätsgesichert (SHA-256-Manifest, Root-Hash, Import-Verifikation mit Pfad-Containment), nicht asymmetrisch signiert.

Feature-Flag — DB-seitiger Schalter je Modul; wird ausschließlich aus der Lizenz-Datei befüllt (License-as-SSoT).

FluentMigrator — .NET-Framework für versionierte Datenbank-Migrationen; läuft bei RPMC automatisch beim Serverstart.

GGUF — Datei-Format für quantisierte lokale KI-Modelle.

HL7 v2 — Nachrichtenstandard im Gesundheitswesen (z. B. ADT-Bewegungsdaten) über MLLP-Transport.

HMAC-SHA256 — Schlüsselabhängiger Hash; verkettet Audit-Einträge und signiert Webhooks.

Hosted Service / Worker — Hintergrunddienst innerhalb des Server-Prozesses für periodische Aufgaben.

Instance-Scope — Mandanten-Isolation: Jede Datenzeile trägt eine `instance_id`; jede Abfrage filtert darauf.

ISiK — Informationstechnische Systeme in Krankenhäusern; FHIR-Basisprofil-Vorgaben der gematik.

JWT — JSON Web Token; signiertes Token-Format für die API-Authentifizierung.

k-Anonymität — Datenschutzprinzip: Auswertungen werden erst ab einer Mindest-Gruppengröße angezeigt.

License-as-SSoT — Die signierte Lizenz-Datei ist die einzige Wahrheitsquelle für aktive Module.

MCP — Model Context Protocol; Werkzeug-Schnittstelle des RPMC-Assistenten (nur Loopback).

MLLP — Minimal Lower Layer Protocol; TCP-Framing für HL7-v2-Nachrichten.

OE — Organisationseinheit (Klinik, Fachabteilung, Station).

OIDC — OpenID Connect; SSO-Standard auf OAuth-2.0-Basis.

Optimistic Locking — Konflikt-Erkennung über Zeilenversion (`row_version`/If-Match) statt Sperren.

ORU / MDM — HL7-Nachrichtentypen für Befunde bzw. Dokumente
geplant · IN ENTW.

Pseudonymisierung — Ersetzen identifizierender Merkmale durch Kennungen; Klartext bleibt im Quellsystem.

RBAC — Role-Based Access Control; rollenbasierte Zugriffssteuerung.

Resilience Queue — Wiederhol-Warteschlange mit exponentiellem Backoff und Dead-Letter-Ablage.

RPKG — Integritätsgeprüftes RPMC-Patch-Paket (SHA-256-Manifest, Root-Hash, Import-Verifikation mit Pfad-Containment, sequenzieller Versions-Guard); Update-Format für den ServerManager.

SAML2 — Security Assertion Markup Language v2; etablierter Enterprise-SSO-Standard.

SCIM 2.0 — System for Cross-domain Identity Management; Standard-Provisioning-Protokoll (Entra ID, Okta).

Schattendatenbank — Lokaler, lesbarer Klon relevanter ORBIS-Tabellen für Betrieb ohne Dauerverbindung zur Oracle-DB.

SIEM — Security Information and Event Management; empfängt RPMC-Events via CEF/Syslog.

SignalR — Echtzeit-Framework von ASP.NET Core (WebSockets mit Fallbacks) für Live-Updates ohne Browser-Refresh.

Stale-Lock-Reaper — Aufräum-Mechanismus, der hängengebliebene Sync-Sperren erkennt und freigibt.

Sub-Sync — Einzeln aktivier- und planbarer Teil-Synchronisationslauf der Direct-DB-Pipeline.

TOTP — Time-based One-Time Password (RFC 6238); zeitbasierte Einmal-Codes für MFA.

Watchfolder — Überwachtes Verzeichnis; abgelegte Dateien werden automatisch importiert.

Write-Guard — Schwellwert-Schutz gegen versehentliche Massenänderungen in Richtung ORBIS.

XRef — Cross-Reference-Tabelle; verknüpft Identitäten über die Quellsysteme (ORBIS-ID, HR-PersNr, AD-Konto, Dienstplan-ID).

---

medivaris Software, Inhaber Karl-Heinz Roth
Hersteller, Anbieter und Lizenzgeber von RPMC · Produktentwicklung: Nikolas Roth
RPMC — Technische Infomappe · Version 2.0 · Stand 22.07.2026