```
module: 37
title: Plattform-Admin
phase: 1
status: detail-spec-draft
last_updated: 2026-05-08
session_source: brainstorming-session-2026-05-07-2206.md
related_lps: [LP-1, LP-6, LP-9, LP-10, LP-13]
spec_style: requirements-pur
```

# Modul 37 — Plattform-Admin

> ***Detail-Spec gemäß LP-9.*** *Eigene Welt für uns als Software-Betreiber.* ***Hardgrenze:*** *läuft auf Sub-Domain* `admin.hivegrid.de`*, ist* ***nicht*** *unter* `/{tenant}/...` *erreichbar, hat eine* ***eigene User-Klasse*** *(* `AdminUser` *) parallel zu* `User` */* `TenantUser`*, eigene Auth-Schicht, eigenes Permission-System. Kein Cross-Routing zwischen App-Welt und Admin-Welt.*

---

## Themenblöcke

```
A. Architektur-Trennung (Sub-Domain, eigene Codebase-Routes, AdminUser-Klasse)
B. Datenmodell (AdminUser, AdminRole, AdminPermission, AdminSession, AdminAuditLog, AdminLead,
AdminDeal, AdminSupportTicket, AdminTenantNote)
C. Auth (Login, 2FA, Session)
D. Permission-System (eigenes Bundle-Modell)
E. Bereiche / Routes
F. Dashboard
G. Tenant-Verwaltung
H. User-Verwaltung
I. Subscription-Management
J. Feature-Flag-Verwaltung (siehe Modul 32)
K. Health/Metrics-Dashboard
L. Cross-Tenant-Audit-Trails
M. Lead-Management (eigenes CRM-Lite für UNSERE Sales)
N. Manuelle Tenant-Anlage und Trial-Verlängerung
O. Support-Bereich
P. Subprozessoren- und AVV-Verwaltung (Querschnitt zu Modul 35)
Q. Klartext-Hinweise
R. Phase 2 Erweiterungen
```

---

## Block A — Architektur-Trennung

### Sub-Domain

```
hivegrid.de             → Marketing-Site (extern, Next.js)
app.hivegrid.de         → SaaS-Anwendung (Tenant-Welt)
admin.hivegrid.de       → Plattform-Admin (uns, dieser Modul-Scope)
```

### Hard-Trennungs-Vorschriften

- **Code-Routes** in `admin.hivegrid.de` sind **nicht** unter `/{tenant}/...` erreichbar — separate Routen-Struktur

- **AdminUser** ist eigene Entität, **nicht** TenantUser, **nicht** User. Kein Login-Pfad zwischen Welten.
- **Cookies/Sessions** sind Domain-getrennt ( `admin.hivegrid.de` setzt nicht das gleiche Cookie wie `app.hivegrid.de` )
- **DB-Zugriff:** AdminUser hat **kein** Tenant-Membership, RLS gilt nicht für admin-eigene Tabellen. Tenant-Daten werden über App-Role mit gesetztem Tenant-Context gelesen (gleicher Pattern wie App-Code, aber Admin kann Tenant-Context wählen). Admin-Code hat **kein** `BYPASSRLS` -Privileg — Admin liest Tenant-Daten bewusst durch RLS, mit explizit gesetztem `app.tenant_id` .
- **Audit-Trennung:** Admin-Aktionen werden in `AdminAuditLog` geloggt (eigene Tabelle), nicht in tenant-internem Audit-Log
- **Permissions:** eigene Tabellen ( `AdminRole` , `AdminPermission` ), unabhängig von Tenant-Rollen

### Zugriff-Pattern

- AdminUser kann auf `admin.hivegrid.de/tenants/:tenantId/...` Tenant-Daten **lesen** (mit Audit)
- Schreib-Aktionen auf Tenant-Daten sind explizit limitiert (siehe Block G — z.B. Status-Wechsel, Trial-Verlängerung, Support-Notes anlegen) — kein freies CRUD auf Tenant-Daten durch Admin
- Direkte Datenbearbeitung im Tenant-Kontext bleibt dem Tenant-User vorbehalten

### Begründung

Maximale Sicherheit (Compromise-Radius eingegrenzt), klare DSGVO-Verantwortlichkeit (Admin-Aktionen separat dokumentiert), klare Permission-Hygiene (Plattform-Rolle nicht versehentlich in Tenant-Welt kopiert).

---

## Block B — Datenmodell

### AdminUser

Plattform-globale Tabelle, **kein** RLS, App-Code-gegated.

- `id` , `created_at` , `updated_at`
- `email` (unique, Pflicht — wird zum Login-Identifier)
- `first_name` , `last_name`
- `password_hash` (Bcrypt, Cost 12)
- `totp_secret` (verschlüsselt; NULL wenn 2FA nicht eingerichtet)
- `totp_enabled` (Boolean — Pflicht für AdminUsers, siehe Block C)
- `totp_backup_codes_hash` (Array, gehashed)
- `last_login_at` , `last_login_ip`
- `status` (Enum: `active | disabled | locked` )
- `disabled_reason` (optional)
- `created_by_admin_user_id` (Bootstrap-Sonderfall: erster Admin via CLI/Seeding)

**Constraints:** `totp_enabled` muss `true` sein bevor `status='active'` möglich ist.

### AdminRole / AdminPermission / AdminUserRole

Eigene RBAC parallel zur Tenant-RBAC.

**AdminRole:**
- `id` , `name` (z.B. *„Super-Admin", „Support", „Sales"*), `description` , `is_template_origin`

**AdminPermission:**
- `id` , `permission_id` (Enum-String, z.B. `admin.tenants.read` )

**AdminRolePermission:**
- `admin_role_id` , `admin_permission_id`

**AdminUserRole:**
- `admin_user_id`, `admin_role_id`

### AdminSession

Server-Side, Postgres.

- `id`, `admin_user_id`, `created_at`, `expires_at`, `last_seen_at`
- `ip`, `user_agent`
- `revoked_at` (für globalen Logout)

### AdminAuditLog

Plattform-globale Audit-Tabelle. Cross-Tenant — auch Aktionen die Tenant-Daten verändern werden hier geloggt.

- `id`, `created_at`
- `admin_user_id` (Akteur, NULL bei System-Cron)
- `actor_kind` (Enum: `admin_user | system | api`)
- `action_type` (z.B. `tenant.status_changed`, `feature_flag.tenant_override.created`, `subscription.manual_override`)
- `target_kind` (Enum: `tenant | tenant_user | admin_user | feature_flag | subscription | subprocessor | avv_version | system`)
- `target_id`
- `summary_text` (Klartext-Pflicht in DE)
- `metadata` (JSONB — Diff, Vorher/Nachher, Reason-Text, etc.)
- `ip`, `user_agent`

**Indizes:** `(admin_user_id, created_at DESC)`, `(target_kind, target_id, created_at DESC)`, `(action_type, created_at DESC)`.

### AdminTenantNote

Frei-Text-Notizen pro Tenant durch Admin (Sales/Support).

- `id`, `tenant_id`, `created_at`, `created_by_admin_user_id`
- `note_text` (Markdown)
- `category` (Enum: `sales | support | technical | billing | other`)
- `pinned` (Boolean — wichtige Notes oben)
- `archived_at`

**Indizes:** `(tenant_id, pinned DESC, created_at DESC)`.

### Lead-Management-Datenmodell (für UNSERE eigenen HiveGrid-Sales)

> ***Wichtig:*** *Bewusst getrennt von Tenant-CRM. Wir verkaufen HiveGrid an Handwerksbetriebe und brauchen ein eigenes Lead-Tool. Variante (a) aus Brainstorming gewählt: separates Lead-Tool im Admin, kein Master-Tenant-Dogfooding.*

**AdminLead:**
- `id`, `created_at`, `updated_at`
- `first_name`, `last_name`, `email`, `phone`, `company_name` (alle nullable — Form-Submissions sind dünn)
- `industry` (Enum: `shk | other`)
- `company_size` (Enum: `1-10 | 11-25 | 26-50 | 51-100 | 100+`)
- `source` (Enum: `marketing_form | demo_booking | referral | manual | inbound_email | partner | other`)
- `source_detail` (Freitext, z.B. *„LinkedIn-Kampagne Mai 2026"*)

- `status` (Enum: `new | qualified | contacted | demo_scheduled | proposal_sent | won | lost | spam`)
- `assigned_to_admin_user_id`
- `lost_reason` (optional Enum + Freitext bei `lost`)
- `notes` (Markdown)
- `converted_to_tenant_id` (FK auf Tenant bei `won` + nach Tenant-Anlage)
- `archived_at`

**AdminDeal** (sehr leichtgewichtig, eigene Sales-Pipeline):
- `id`, `created_at`, `updated_at`
- `admin_lead_id` (FK)
- `expected_arr_eur` (geschätzter Jahresumsatz)
- `seats_office_estimated`, `seats_field_estimated`
- `stage` (Enum: `discovery | demo | proposal | negotiation | won | lost`)
- `expected_close_date`
- `assigned_to_admin_user_id`
- `notes`

**AdminLeadActivity** (eigener Activity-Stream für Lead-Management):
- `id`, `admin_lead_id`, `created_at`, `actor_admin_user_id`
- `activity_type` (Enum: `email_sent | email_received | call_made | demo_held | proposal_sent | note_added | stage_changed | assigned_changed`)
- `summary_text` (Pflicht, DE)
- `metadata` (JSONB)

### AdminSupportTicket

Inbound-Feedback aus Tenants und In-App-Feedback-Widget-Submissions.

- `id`, `created_at`, `updated_at`
- `source_kind` (Enum: `in_app_feedback_widget | inbound_email | manual`)
- `tenant_id` (NULL wenn nicht zuordenbar)
- `tenant_user_id` (NULL wenn anonym)
- `subject`, `body` (Markdown)
- `status` (Enum: `open | in_progress | waiting_user | resolved | closed`)
- `priority` (Enum: `low | normal | high | urgent`)
- `assigned_to_admin_user_id`
- `tags[]` (frei: `bug`, `feature_request`, `billing`, `onboarding`, etc.)
- `resolved_at`

**AdminSupportTicketMessage** (Konversation):
- `id`, `ticket_id`, `created_at`, `actor_kind` (Enum: `admin_user | tenant_user | system`), `actor_user_id`, `body`, `attachments[]`

---

## Block C — Auth

### Login

- Email + Passwort (NIST-konform analog Tenant-User: ≥12 Zeichen, HIBP-Check, Bcrypt Cost 12)
- **2FA Pflicht** für jeden AdminUser (kein Opt-Out — Plattform-Sicherheit)
- TOTP-only, gleicher Flow wie Tenant-User-2FA
- Magic-Link **nicht** verfügbar für AdminUser (zu hohes Risiko bei Email-Account-Compromise)

### Login-Schutz

- Rate-Limit: 5 Fehlversuche / 15 Min → 15 Min Lockout
- Captcha ab 3. Fehlversuch
- Bei 10 Fehlversuchen in 24h: AdminUser-Status auf `locked`, manuelle Reaktivierung durch anderen Super-Admin

### Session

- Server-Side, Postgres
- Default-Lebenszeit: **8 Stunden** (kürzer als Tenant-User wegen höherem Risiko)
- Inaktivitäts-Timeout: 1h ohne Aktivität → Re-Login
- IP-Bindung: bei IP-Wechsel mitten in Session → Re-Login
- Cookie: HTTP-only, Secure, SameSite=Strict

### Session-Übersicht

- Eigene Settings-Seite `admin.hivegrid.de/settings/security/`
- Liste aktiver Sessions, individueller Revoke
- Trust-Device **nicht** verfügbar (jeder Login mit 2FA)

### Initial-AdminUser

- Bootstrap via CLI-Seed-Script (kein Public-Signup)
- Erste AdminUser-Rolle: `super_admin` (alle Permissions)
- Weitere AdminUsers werden durch `super_admin` über UI angelegt

---

## Block D — Permission-System

### Bundle-Stufen pro Bereich (analog Variante 2 in Modul 27)

```
admin.tenants.none / read / write / admin
admin.users.none / read / write / admin
admin.subscriptions.none / read / write / admin
admin.feature_flags.none / viewer / operator / owner   (siehe Modul 32)
admin.metrics.none / read
admin.audit.none / read
admin.leads.none / read_own / read_all / write / admin
admin.support.none / read / write / admin
admin.legal.none / read / write   (Subprozessoren, AVV-Versionen)
admin.admin_users.none / read / write / admin   (AdminUser-Verwaltung)
admin.system.none / read / admin   (Infra-Aktionen, Worker-Steuerung)
```

### Default-Rollen (Seeding)

| Rolle | Bundles |
|---|---|
| `super_admin` | alle Permissions, alle Bundles auf höchster Stufe |
| `support` | `tenants.read`, `users.read`, `subscriptions.read`, `metrics.read`, `audit.read`, `support.admin`, `feature_flags.viewer`, `legal.read` |
| `sales` | `leads.admin`, `tenants.read`, `subscriptions.read`, `metrics.read`, `support.read` |
| `engineering` | `metrics.read`, `audit.read`, `feature_flags.owner`, `system.admin`, `tenants.read`, `users.read` |
| `viewer` (read-only) | `tenants.read`, `users.read`, `subscriptions.read`, `metrics.read`, `audit.read`, `feature_flags.viewer`, `support.read`, `leads.read_all`, `legal.read` |

**Anpassbarkeit:** alle Rollen editierbar wie in Tenant-RBAC. Rollen sind Pre-Sets, atomare Permissions kombinierbar.

### Self-Permission-Schutz

- Letzter `super_admin` kann sich nicht selbst entziehen
- AdminUser kann sich nicht selbst deaktivieren

---

## Block E — Bereiche / Routes

```
admin.hivegrid.de/                          → Login
admin.hivegrid.de/login                     → Login-Form
admin.hivegrid.de/2fa                       → 2FA-Eingabe
admin.hivegrid.de/dashboard                 → Plattform-Dashboard (siehe Block F)
admin.hivegrid.de/tenants                   → Tenant-Liste
admin.hivegrid.de/tenants/new               → Manuelle Tenant-Anlage (siehe Block N)
admin.hivegrid.de/tenants/:tenantId         → Tenant-Detail (siehe Block G)
admin.hivegrid.de/tenants/:tenantId/...     → Tenant-Sub-Bereiche
admin.hivegrid.de/users                     → User-Verwaltung (siehe Block H)
admin.hivegrid.de/users/:userId             → User-Detail
admin.hivegrid.de/admin-users               → AdminUser-Verwaltung
admin.hivegrid.de/admin-users/:id           → AdminUser-Detail (Rollen, Audit, Sessions)
admin.hivegrid.de/admin-users/roles         → AdminRole-Editor
admin.hivegrid.de/subscriptions             → Subscription-Management (siehe Block I)
admin.hivegrid.de/subscriptions/:id         → Subscription-Detail
admin.hivegrid.de/feature-flags/            → siehe Modul 32
admin.hivegrid.de/health                    → Health/Metrics-Dashboard (siehe Block K)
admin.hivegrid.de/health/workers            → Worker-Queue-Detail
admin.hivegrid.de/health/kwp-sync           → KWP-Sync-Status pro Tenant
admin.hivegrid.de/audit                     → Cross-Tenant-Audit-Trails (siehe Block L)
admin.hivegrid.de/leads                     → Lead-Inbox (siehe Block M)
admin.hivegrid.de/leads/:id                 → Lead-Detail
admin.hivegrid.de/leads/deals               → Deal-Pipeline (Sales)
admin.hivegrid.de/leads/deals/:id           → Deal-Detail
admin.hivegrid.de/support                   → Support-Inbox (siehe Block O)
admin.hivegrid.de/support/:ticketId         → Ticket-Detail
admin.hivegrid.de/legal/subprocessors       → Subprozessoren-Verwaltung (siehe Block P)
admin.hivegrid.de/legal/avv                 → AVV-Versions-Verwaltung
admin.hivegrid.de/settings/account          → Eigenes Profil
admin.hivegrid.de/settings/security         → 2FA, Sessions
```

**Sidebar-Hauptpunkte:** Dashboard, Tenants, Users, Subscriptions, Feature-Flags, Health, Audit, Leads, Support, Legal, AdminUsers, Settings.

---

# Block F — Dashboard

## Zu zeigende Daten

- **Tenants-Übersicht:**
- Gesamt-Tenant-Zahl mit Trend (vs. Vormonat)
- Aktive Tenants nach Status (`trial`, `active`, `past_due`, `frozen`, `suspended`, `deletion_pending`)
- Neue Tenants letzte 7/30 Tage
- **MRR / Wachstum:**
- Monthly Recurring Revenue aktuell
- MRR-Trend (Chart letzte 12 Monate)
- Neue MRR / Churned MRR / Expansion MRR letzten Monat
- ARPU (Average Revenue per Tenant)
- **Health:**
- System-Status-Indikator (grün/gelb/rot mit Klartext bei Auffälligkeit)
- Worker-Queue-Backlog (Anzahl Jobs, oldest pending)
- Error-Rate letzte 24h
- KWP-Sync-Health (Anzahl Tenants mit aktivem Heartbeat / fehlerhaft)
- **Aktivität:**
- Anzahl aktiver Tenant-User letzte 24h / 7d
- Top 5 Tenants nach Activity
- **Sales-Pipeline (UNSERE):**
- Anzahl Leads in jedem Status
- Pipeline-Volumen (Summe `expected_arr_eur` pro Stage)
- **Support:**
- Offene Tickets, ältester offener
- Durchschnittliche Lösungs-Zeit

## Benötigte UI-Elemente

- Widget-basierte Anzeige (Kacheln pro Bereich)
- Klick auf jede Kachel → Drill-Down auf zugehörigen Bereich
- Zeitraum-Wechsler (24h / 7d / 30d / 12 Monate)
- Refresh-Funktion

## Performance

- Daten werden gecacht (Redis, TTL 5 Min)
- Live-Werte (System-Status, Queue-Backlog) sind direkter DB-/Redis-Read
- Aggregat-Berechnung als Background-Job (Hourly), Dashboard liest Snapshot

---

# Block G — Tenant-Verwaltung

## Tenant-Liste ( `admin.hivegrid.de/tenants` )

**Zu zeigende Daten pro Eintrag:**
- Tenant-Name + Slug
- Status mit Klartext-Farbe
- Subscription-Plan + MRR
- Erstellt-Datum
- Anzahl Tenant-User
- Letzte User-Aktivität
- Branche, Größe
- Verantwortlicher AdminUser (für Sales-Zuordnung)

**Benötigte UI-Elemente:**
- Tabellarische Liste
- Filter: Status, Plan, Branche, Größe, Erstellt zwischen, mit/ohne aktive Subscription
- Suche über Name + Slug + Email der Tenant-Admins
- Sortierung pro Spalte
- Spalten-Konfiguration
- Bulk-Actions: Tag setzen, an AdminUser zuweisen, Email-Versand (Marketing-Newsletter, Pflicht-Hinweise)

**Operationen:**
- Klick auf Tenant → Detail
- Schnelle Status-Aktion: Suspend / Reaktivieren (mit Bestätigung)
- Manuelle Tenant-Anlage über Add-Funktion (siehe Block N)

## Tenant-Detail ( `admin.hivegrid.de/tenants/:tenantId` )

**Zu zeigende Daten:**
- Stammdaten: Slug, Name, Branche, Größe, Erstellt-Datum, Status, Subscription, primärer Tenant-Admin
- Kontakt-Info (Email, Telefon, Adresse aus Signup)
- Tenant-Admins-Liste (welche User haben `tenant.admin` )
- Activity-Indikatoren: letzte 7/30 Tage, MAU, Speicher-Nutzung, KWP-Sync-Status
- Subscription-Detail: Plan, Seats, MRR, nächste Rechnung, Stripe-Customer-ID (mit Link zu Stripe-Dashboard)
- Trial-Info (falls Trial): Restdauer, manuelle Verlängerung möglich
- Feature-Flag-Auswertungen für diesen Tenant (siehe Modul 32)
- Subprozessoren-Status (welche AVV-Version akzeptiert)
- Notizen (Liste der `AdminTenantNote` mit Pin-Funktion)
- Activity-Log (Cross-Modul: Logins, Status-Wechsel, Plan-Wechsel, Support-Interaktionen — Quelle: AdminAuditLog gefiltert auf `target_kind=tenant, target_id=tenantId` + Tenant-eigenes Audit für Lese-Sicht)

**Benötigte UI-Elemente:**
- Sektionierte Anzeige: Stammdaten, Subscription, Users, Activity, Notes, Audit, Feature-Flags, Health
- Tab- oder Sektionen-Navigation
- Notes-Editor (Markdown, Kategorie, Pin-Toggle)
- Status-Wechsler mit Bestätigungs-Dialog und Reason-Text-Pflicht
- Trial-Verlängerungs-Funktion: Datum-Picker + Reason-Text
- Subscription-Override-Funktion (siehe Block I)
- Impersonate-Funktion: *„Als Tenant-Admin einloggen"* — siehe unten

**Operationen:**
- Status ändern: → `suspended` , → `active` , → `frozen` mit Klartext-Reason
- Trial verlängern (Datum-Eingabe + Reason)
- Subscription manuell überschreiben (Plan, Seats, Sonderpreis)

- Tenant-Notiz anlegen/editieren/pinnen
- Feature-Flag-Override für diesen Tenant setzen
- Email an Tenant-Admin(s) senden (Template-basiert oder freier Text)
- Tenant-Daten-Export anstoßen (für Support-Fall, vermittelt Tenant-Admin den Download — siehe Modul 35)
- Tenant-Lösch-Workflow im Auftrag des Tenant-Admins anstoßen ( `admin_initiated` -Reason, siehe Modul 35)
- KWP-Sync-Status detailliert ansehen mit Re-Trigger-Funktion

### Impersonate-Funktion

- AdminUser kann *„Als Tenant-User einloggen"* klicken
- Reason-Text-Pflicht (z.B. *„Bug-Reproduktion Ticket #123"*)
- Generiert temporäre Session als gewählter Tenant-User (max. 30 Min)
- Sichtbar im Tenant für andere Tenant-User: Klartext-Banner *„Plattform-Support hat sich gerade als [Name] eingeloggt — Ticket #123"*
- Audit-Log mit voller Reason-Spur
- Permission: `admin.tenants.admin` + spezielles `admin.tenants.impersonate` Bundle

### Permissions

- Liste/Detail: `admin.tenants.read`
- Notes anlegen/bearbeiten: `admin.tenants.write`
- Status-Wechsel, Trial-Verlängerung, Subscription-Override, Lösch-Workflow: `admin.tenants.admin`
- Impersonate: `admin.tenants.impersonate`
- Klartext-Reason-Text bei sensitiven Aktionen Pflicht

---

## Block H — User-Verwaltung

### User-Liste ( `admin.hivegrid.de/users` )

**Zu zeigende Daten pro Eintrag:**
- Email, Name
- Anzahl Tenant-Memberships
- Letzte Aktivität
- 2FA-Status
- Erstellt-Datum
- Status (aktiv / disabled / gelöscht)

**Benötigte UI-Elemente:**
- Tabellarische Liste mit Filter (Status, 2FA, mit/ohne Memberships)
- Suche über Email + Name
- Klick → Detail

**Operationen:**
- Detail öffnen
- Bulk-Email an User-Auswahl (z.B. Pflicht-Mitteilungen)

### User-Detail ( `admin.hivegrid.de/users/:userId` )

**Zu zeigende Daten:**
- Stammdaten (Email, Name, Sprache, 2FA-Status, OAuth-Verknüpfungen)
- Liste aller Tenant-Memberships mit Rolle pro Tenant
- Aktive Sessions
- Letzte Logins (Datum, IP, Geo, User-Agent)

- 2FA-Trust-Devices-Liste
- Audit-Spur (User-bezogen)
- Account-Lösch-Status (falls in Bearbeitung)

**Operationen:**
- 2FA zurücksetzen (mit Reason-Text-Pflicht)
- Sessions revoken
- Email ändern (mit Audit + Verifikations-Email)
- User deaktivieren / reaktivieren
- Account-Lösch-Funktion im Auftrag (für Support-Fälle; siehe Modul 35)
- Wiederherstellung innerhalb 24h-Reverse-Window (siehe Modul 27 D.3)

**Permissions:** `admin.users.read` für Lese, `admin.users.write` für Stamm, `admin.users.admin` für Lösch-/2FA-Reset.

### AdminUser-Verwaltung (`admin.hivegrid.de/admin-users`)

**Zu zeigende Daten pro Eintrag:**
- Email, Name, Rollen, Status, 2FA-Status, letzter Login, Anzahl aktiver Sessions

**Benötigte UI-Elemente:**
- Liste mit Filter / Suche
- AdminUser-Anlegen (Email, Name, Initial-Rolle → Magic-Link-Setup-Email mit 2FA-Setup-Pflicht beim ersten Login)
- AdminRole-Editor (analog Tenant-RBAC)

**Operationen:**
- AdminUser anlegen / deaktivieren / Rolle ändern
- 2FA zurücksetzen (durch anderen Super-Admin)
- Sessions revoken
- AdminRoles editieren

**Permissions:** `admin.admin_users.admin`. Self-Permission-Schutz für letzten `super_admin`.

---

## Block I — Subscription-Management

### Liste (`admin.hivegrid.de/subscriptions`)

**Zu zeigende Daten pro Eintrag:**
- Tenant-Name, Plan, MRR, Status (active, trialing, past_due, canceled), Renewal-Datum, Stripe-Sync-Status

**Benötigte UI-Elemente:**
- Tabellarische Liste mit Filter (Plan, Status, MRR-Range)
- Suche über Tenant-Name
- Klick → Detail

### Detail (`admin.hivegrid.de/subscriptions/:id`)

**Zu zeigende Daten:**
- Stripe-Customer-ID, Subscription-ID (mit Link zu Stripe-Dashboard)
- Aktueller Plan, Seats (Office + Field), MRR
- Trial-Status, Trial-Ende
- Rechnungs-History
- Override-Vermerke (manueller Plan / Sonderpreis)

**Operationen:**
- **Stripe-Sync forcieren** (zieht aktuellen Stripe-Stand und gleicht ab)

- **Manuelle Override** für Enterprise-Kunden:
- Plan überschreiben
- Seats-Limit überschreiben
- Sonderpreis (Custom-MRR)
- Reason-Text-Pflicht
- Override-Eintrag im Audit
- Klartext-Hinweis im Tenant-Detail: *„Manuelle Konditionen aktiv — siehe Override-Vermerk*"
- **Trial verlängern** (analog Block G)
- **Manuelle Rechnung anstoßen** (für Setup-Fees, Custom-Onboarding)
- **Abo kündigen** (Stripe-Cancel + Tenant-Lösch-Workflow-Trigger)
- **Refund anstoßen** (mit Reason-Text + Stripe-Sync)

**Permissions:** Lese `admin.subscriptions.read`, Schreib-Aktionen `admin.subscriptions.admin`.

### Stripe-Webhook-Sync

- Events von Stripe (Subscription-Update, Payment-Failed, Invoice-Paid) werden automatisch in Tenant-Daten gespiegelt (Modul 28)
- Manuelle Re-Sync-Funktion in diesem Bereich, falls Webhook ausgefallen war

---

## Block J — Feature-Flag-Verwaltung

**Komplette Spec siehe Modul 32.** Nur Querverweise hier:

- Routes leben unter `admin.hivegrid.de/feature-flags/`
- Permission-Bundles `admin.feature_flags.viewer | operator | owner`
- Audit fließt in `AdminAuditLog`

---

## Block K — Health/Metrics-Dashboard

### System-Status

**Zu zeigende Daten:**
- Web-Service: Uptime, P50/P95-Latency, Error-Rate
- API-Service: gleiches
- Worker-Service: Backlog, oldest-pending, Error-Rate
- Scheduler-Service: letzter Run pro Cron-Job
- DB: Connection-Pool-Status, slow-Query-Count
- Redis: Memory, Connection-Count
- Storage (Scaleway): aktuelle Nutzung pro Bucket-Pfad

**Benötigte UI-Elemente:**
- Real-Time-Updates (Polling oder Websocket)
- Klick auf Service → Detail mit Metriken-Charts (letzte 24h / 7d)
- Alert-Indikator bei Auffälligkeiten

### Worker-Queues (`admin.hivegrid.de/health/workers`)

- Liste aller BullMQ-Queues (Automation, Email-Sending, KWP-Sync, GdprExport, etc.)
- Pro Queue: Active, Waiting, Delayed, Failed, Completed
- Klick auf Queue → Job-Liste mit Detail (Payload, Error-Stack)
- Funktionen: Retry failed Jobs (Bulk + einzeln), Pause/Resume Queue, Job manuell aus Queue entfernen

### KWP-Sync-Status ( `admin.hivegrid.de/health/kwp-sync` )

- Liste aller Tenants mit aktiver KWP-Integration
- Pro Tenant: Heartbeat-Status (grün/gelb/rot, letzter Heartbeat-Zeitstempel), Queue-Backlog, letzter erfolgreicher Sync, Error-Count letzte 24h
- Klick auf Tenant → Detail mit kwpsync.exe-Logs, Versions-Info, Schema-Drift-Sentinel-Hits

### Error-Rate

- Sentry-Integration (oder direkter Read aus DB)
- Top 10 Errors letzten 24h mit Count und Klick-Drill auf Stack-Trace

**Permissions:** `admin.metrics.read` für Lese, `admin.system.admin` für Worker-Steuerung.

---

## Block L — Cross-Tenant-Audit-Trails

### Audit-Log ( `admin.hivegrid.de/audit` )

**Zu zeigende Daten:**
- Cross-Tenant-Liste aus `AdminAuditLog` + aggregierter Read-Zugriff auf alle Tenant-internen Audit-Logs
- Pro Eintrag: Datum, Akteur (AdminUser oder Tenant-User), Tenant-Kontext, Action-Type, Summary-Text, Diff (in Detail-Drawer)

**Benötigte UI-Elemente:**
- Tabellarische Liste mit Filter:
- Akteur (AdminUser, Tenant-User, System)
- Tenant (Auto-Complete)
- Action-Type (Multi-Select aus bekannten Typen)
- Zeitraum (Date-Range)
- Target-Kind
- Volltext-Suche in `summary_text`
- Detail-Drawer mit voller Diff
- Export (CSV/JSON) für Compliance-Anfragen

**Operationen:**
- Audit-Einträge sind **read-only** — keine Edit-/Delete-Funktion
- Retention: 24 Monate Default für AdminAuditLog (AV-aligned mit Modul 21)

**Permissions:** `admin.audit.read` für alle Lese-Zugriffe. Compliance-relevante Exports erfordern `admin.audit.admin`.

---

## Block M — Lead-Management (UNSERE Sales)

### Lead-Inbox ( `admin.hivegrid.de/leads` )

**Zu zeigende Daten pro Eintrag:**
- Name, Email, Firma, Branche, Größe
- Source mit Detail
- Status mit Farbe
- Verantwortlicher AdminUser
- Letzte Activity-Datum
- Erstellt-Datum

**Benötigte UI-Elemente:**
- Tabellarische Liste
- Filter: Status, Source, Branche, Größe, Verantwortlicher, Erstellt zwischen
- Suche über Name + Email + Firma
- Sortierung pro Spalte
- Bulk-Actions: Status setzen, Verantwortlichen zuweisen, Tag setzen, exportieren

**Operationen:**
- Add-Funktion (manuell anlegen)
- Detail öffnen

## Lead-Detail (`admin.hivegrid.de/leads/:id`)

**Zu zeigende Daten:**
- Stammdaten (alle Lead-Felder)
- Activity-Stream (`AdminLeadActivity`)
- Verknüpfter Deal falls vorhanden (`AdminDeal`)
- Verknüpfter Tenant falls `won` und nach Tenant-Anlage
- Notes (Markdown, kommen aus den `note_added`-Activities)

**Benötigte UI-Elemente:**
- Sektionen: Stamm, Activities, Deal, Notes
- Inline-Edit für Stammdaten
- Activity-Add-Funktionen: Note, Email-Log, Anruf-Log, Demo-Log
- Status-Wechsler
- Promote-to-Tenant-Funktion bei `won`: triggert manuelle Tenant-Anlage (siehe Block N) und setzt `converted_to_tenant_id`

**Operationen:**
- Stammdaten editieren
- Status wechseln
- Verantwortlichen ändern
- Notes/Activities anlegen
- Deal anlegen / verknüpfen (öffnet Modal mit Deal-Stammdaten)
- Promote zu Tenant
- Hard-Delete (nur als `spam` markiert; sonst archivieren)

## Deal-Pipeline (`admin.hivegrid.de/leads/deals`)

**Zu zeigende Daten:**
- Kanban-View mit Stage-Spalten (`discovery | demo | proposal | negotiation | won | lost`)
- Pro Karte: Lead-Name, Firma, ARR, Verantwortlicher, Expected-Close-Date

**Benötigte UI-Elemente:**
- Kanban-Layout (Stages als Spalten)
- Drag&Drop zwischen Stages
- Filter: Verantwortlicher, ARR-Range, Expected-Close-Date
- Detail-Drawer beim Klick

**Operationen:**
- Drag&Drop zum Stage-Wechsel
- Klick → Deal-Detail mit Verknüpfung zu `AdminLead`

## Permissions

- `admin.leads.read_own` (Sales sieht seine eigenen Leads)
- `admin.leads.read_all` (Sales-Leitung)

- `admin.leads.write` (anlegen, editieren)
- `admin.leads.admin` (Bulk-Operations, Hard-Delete)

---

## Block N — Manuelle Tenant-Anlage und Trial-Verlängerung

### Manuelle Tenant-Anlage (`admin.hivegrid.de/tenants/new`)

**Use-Cases:**
- Demo-Termin: Sales legt Tenant für Demo-Vorbereitung an
- Enterprise-Onboarding: nach Vertrag wird Tenant manuell angelegt mit Sonder-Konditionen
- Lead-Promotion: aus `AdminLead` nach `won`

**Benötigte UI-Elemente:**
- Form mit Feldern:
- Tenant-Name + Slug (Live-Validation analog Modul 27 A.2)
- Branche, Größe
- Initial-Tenant-Admin: Email, Vor-/Nachname (kann existierender User sein, dann Membership; sonst neue Account-Erstellung mit Magic-Link-Setup-Email)
- Initial-Subscription: Plan auswählen (inkl. Custom-Override-Felder für Enterprise)
- Trial-Dauer (Default 7 Tage, anpassbar)
- Optional: Demo-Daten provisionieren (Häkchen pro Demo-Set: SHK-Demo, leer)
- Reason-Text (Pflicht — was ist der Grund für die manuelle Anlage)

**Operationen:**
- Validierung aller Felder
- Bestätigungs-Klick → Tenant wird angelegt, Initial-Tenant-Admin bekommt Email mit Magic-Link
- AdminAuditLog-Eintrag mit Reason
- Optional: Verknüpfung zu `AdminLead` (`converted_to_tenant_id` setzen)

### Trial-Verlängerung

- Aus Tenant-Detail-View aufrufbar (siehe Block G)
- Form: neues Trial-Ende-Datum + Reason-Text
- Bestätigungs-Klick → Tenant-Status bleibt `trial`, `trial_ends_at` wird aktualisiert
- Audit-Eintrag mit Vorher/Nachher-Datum + Reason
- Klartext-Hinweis: *„Trial wurde bis [Datum] verlängert von [AdminUser] mit Reason: [Reason]"*

**Permissions:** `admin.tenants.write` für Verlängerung, `admin.tenants.admin` für manuelle Anlage und Subscription-Override beim Anlegen.

---

## Block O — Support-Bereich

### Inbound-Quellen

- **In-App-Feedback-Widget-Submissions:** Tenant-User klickt im Workspace auf Feedback-Button (siehe Modul 39) → erzeugt `AdminSupportTicket` mit `source_kind=in_app_feedback_widget`
- **Inbound-Email:** Email an `support@hivegrid.de` wird parsed, erzeugt Ticket mit `source_kind=inbound_email`. Tenant-Zuordnung via Sender-Email-Match auf User-Tabelle.
- **Manuelle Anlage:** AdminUser legt Ticket im Auftrag eines Anrufs/Chats an (`source_kind=manual`)

### Support-Inbox (`admin.hivegrid.de/support`)

**Zu zeigende Daten pro Eintrag:**
- Subject, Tenant-Name (falls zuordenbar), Tenant-User, Source, Status, Priority, Tags, Assigned, Erstellt-Datum, letzte Activity

**Benötigte UI-Elemente:**
- Tabellarische Liste
- Filter: Status, Priority, Tags, Assigned, Source, Tenant
- Suche über Subject + Body
- Sortierung
- Bulk-Actions: Status setzen, zuweisen, taggen, schließen

**Operationen:**
- Klick → Detail
- Quick-Assign

### Ticket-Detail (`admin.hivegrid.de/support/:ticketId`)

**Zu zeigende Daten:**
- Stammdaten
- Konversation (alle `AdminSupportTicketMessage`s)
- Verlinkung auf Tenant + User (Klick öffnet Tenant/User-Detail)
- Activity-Log (Status-Änderungen, Zuweisungen)

**Benötigte UI-Elemente:**
- Konversations-Anzeige chronologisch
- Reply-Editor (Markdown + Anhänge)
- Status-Wechsler, Priority-Wechsler, Tag-Editor, Assignee-Wechsler
- Reaktions-Templates (vorgefertigte Antworten)
- Internal-Note-Funktion (Notiz nur für AdminUsers, nicht zum Tenant gesendet)

**Operationen:**
- Antwort senden (Reply geht als Email an Tenant-User-Email-Adresse + erscheint im Tenant ggf. als In-App-Notification)
- Status wechseln
- Ticket schließen mit Resolution-Note
- Re-Open

**Permissions:** `admin.support.read` für Lese, `admin.support.write` für Antworten, `admin.support.admin` für Bulk + Lösch.

---

## Block P — Subprozessoren- und AVV-Verwaltung

**Querschnitt zu Modul 35.**

### Subprozessoren-Verwaltung (`admin.hivegrid.de/legal/subprocessors`)

**Zu zeigende Daten:**
- Liste aller `Subprocessor`-Einträge (siehe Modul 35 Block A)
- Pro Eintrag: alle Felder + Klartext über Aktivitäts-Datum

**Benötigte UI-Elemente:**
- Tabellarische Liste mit Filter (EU/Drittland, Datenkategorie)
- Add/Edit-Funktionen
- Archivieren

**Operationen:**
- Subprozessor anlegen / editieren / archivieren
- Bei Anlage / Änderung: Notification-Trigger (siehe Modul 35 Block H) — Email an alle Tenant-Admins mit 14-Tage-Widerspruchs-Frist
- Audit-Eintrag

### AVV-Versions-Verwaltung (`admin.hivegrid.de/legal/avv`)

**Zu zeigende Daten:**
- Liste aller AVV-Versionen mit Aktiv-ab-Datum
- Pro Version: Anzahl akzeptierender Tenants, Anzahl ausstehender Re-Akzeptanzen
- PDF-Vorlage-Editor

**Operationen:**
- Neue AVV-Version anlegen (PDF-Vorlage hochladen, Aktiv-ab-Datum wählen, Klartext-Beschreibung der Änderungen)
- Re-Akzeptanz-Trigger für alle Tenants
- Klartext über betroffene Tenants
- Audit-Eintrag

**Permissions:** `admin.legal.read` und `admin.legal.write`.

---

## Block Q — Klartext-Hinweise / -Fehler

- Tenant-Status-Wechsel ohne Reason-Text: Block mit Klartext *„Bitte Reason-Text angeben — wird im Audit-Log dauerhaft gespeichert."*
- Impersonate ohne Reason: Block analog
- Subscription-Override ohne Bestätigung: Modal *„Diese Konditionen weichen vom Standard-Plan ab. Stripe wird nicht angepasst — manuelle Rechnungs-Erstellung erforderlich. [Verstanden, fortfahren]"*
- AdminUser-Anlegen mit existierender Email: Klartext *„Diese Email gehört bereits zu einem AdminUser ‚X' — bitte direkt diesen verwalten."*
- Self-Permission-Schutz: *„Du bist der einzige Super-Admin. Permission-Entzug nicht möglich. Bitte zuerst weiteren Super-Admin anlegen."*
- AdminUser-Lockout durch zu viele Fehlversuche: Klartext *„Account gesperrt nach 10 Fehlversuchen. Bitte anderen Super-Admin um Reaktivierung bitten."*
- Tenant-Lösch-Anstoß im Auftrag: *„Du startest die Kündigung für Tenant ‚X' im Auftrag. Bestätigung-Email geht an alle Tenant-Admins mit Cancel-Link."*
- Stripe-Sync-Konflikt: *„Stripe-Stand und HiveGrid-Stand weichen ab. Bitte manuell prüfen: [Stripe-Wert] vs. [HiveGrid-Wert]."*
- Worker-Queue-Pause: *„Queue ‚X' pausiert. Neue Jobs werden eingereiht aber nicht verarbeitet. Vergiss nicht zu reaktivieren!"*
- Cross-Tenant-Email-Bulk: *„Diese Email geht an X Tenant-Admins. Bitte nochmal Subject + Body prüfen. [Senden]"*
- Lead-Promote-to-Tenant: *„Tenant wird angelegt mit Initial-Admin [Email]. Bitte Subscription + Trial-Dauer wählen vor Anlage."*

---

## Block R — Phase 2 / spätere Features

- **AI-Generations-Logs sichtbar** — alle KI-Aufrufe pro Tenant (Token-Verbrauch, Modell, Kosten) für Cost-Analyse + Fehler-Debug
- **KB-Verwaltung global** — zentrale Help-Center-Artikel-Pflege (siehe Modul 38)

- **Help-Desk-Backend mit Tickets** — voller Helpdesk-Workflow (SLA, Eskalations-Stufen, Customer-Satisfaction-Surveys)
- **Marketing-Tools** — Tenant-Segmentierung für Marketing-Newsletter aus Admin heraus, Marketing-Automations für Trial-Conversion
- **Custom-Domain-Verwaltung** — Enterprise-Tenants mit eigenen Domains (siehe Modul 27 A.3 Phase 2)
- **Cohort-Analyse** — Retention, Churn-Predictoren, Feature-Adoption pro Cohort
- **Plattform-API** — programmatischer Zugriff auf Admin-Funktionen (für interne Automation, z.B. Lead → Tenant via Make/n8n)
- **Cross-Tenant-Bulk-Migration** — bei Schema-Änderung Migration über alle Tenants koordinieren
- **Tenant-zu-Tenant-Daten-Move** — Edge-Case für Firmen-Restrukturierung (Konzern-Aufteilung)
- **Erweiterte AdminUser-2FA-Optionen** — Hardware-Keys (WebAuthn/FIDO2)
- **Drittland-Audit-Log und DSFA-Generator** (siehe Modul 35 Phase 2)
- **Marketing-Site-CMS** (falls Marketing-Site auf eigene Plattform migriert wird)

---

## Block S — Architektur-Vorschriften (Querschnitt)

- **LP-1 — Eine Handwerksfirma = ein Tenant:** Hardgrenze Admin-Welt vs. App-Welt (Block A)
- **LP-6 — Anti-Agentur-Positionierung:** Plattform-Admin ist UNSER Werkzeug, kein Tenant-Tool. Klare Trennung wahrt Selbst-Wartbarkeits-Anspruch der Tenants
- **LP-9 — Spec-Tiefe:** keine UI-Layouts
- **LP-10 — Klartext-Diagnostik:** alle sensitiven Aktionen mit Reason-Text + Klartext-Bestätigung (Block Q)
- **AV #6 — Audit-Trail-Default:** AdminAuditLog erfasst alle Admin-Aktionen
- **AV #20 — Action-Boundary:** keine KI-getriebenen schreibenden Aktionen ohne explizite AdminUser-Bestätigung

---

## Detail-Spec-Status: DRAFT

Erste vollständige Detail-Spec für Modul 37. Bewusst breit, da Modul ein Hub für mehrere Querschnitts-Themen (Feature-Flags M32, DSGVO M35, Subscriptions M28) ist.

**Offene Detail-Punkte (nicht-blockierend):**
- Konkrete AdminPermission-IDs final mit Modulen 32 und 35 abstimmen
- Impersonate-Mechanik technisch ausarbeiten (Cookie-Scope, Banner-Anzeige im App-Workspace)
- Inbound-Email-Parser für Support-Tickets (Provider-Wahl)
- Stripe-Webhook-Sync-Detail in Modul 28
- Worker-Queue-Steuerungs-API (BullMQ-spezifisch)
- AVV-PDF-Templating (Template-Engine-Wahl)
- Subframe-UI-Designs separat
- Initial-AdminUser-Bootstrap-Skript (Implementierungs-Detail)