```
module: 39
title: Help-Desk / Support
phase: 1
status: detail-spec-draft
last_updated: 2026-05-08
session_source: brainstorming-session-2026-05-07-2206.md
related_lps: [LP-6, LP-9, LP-10]
spec_style: requirements-pur
```

Modul 39 — Help-Desk / Support

> ***Detail-Spec gemäß LP-9.*** *Phase 1 basic: Support-Email-Mailbox sichtbar im Plattform-Admin (Modul 37) und ein einfaches In-App-Feedback-Widget („Problem melden"), dessen Submissions im Plattform-Admin landen. Kein Ticket-System, kein Live-Chat, kein User-facing-Status-Tracking in Phase 1.*

Themenblöcke

```
A. Datenmodell (Feedback)
B. In-App-Feedback-Widget (Tenant-Seite)
C. Support-Mailbox (Plattform-Admin-Seite)
D. Plattform-Admin-Feedback-Liste
E. Permissions
F. Phase 2
```

Block A — Datenmodell

A.1 — Feedback (zentrale Tabelle)

Tabelle: `Feedback`. **Liegt im Plattform-Admin-Datenraum** (kein RLS auf Tenant-Ebene; siehe Modul 27 A.4 — Sonderfall „kein Tenant-RLS, App-Code permission-gegated"). `tenant_id` und `user_id` sind referenziell vorhanden, aber Lese-Zugriff über Plattform-Admin-Permissions, nicht über Tenant-Permissions.

Felder:
- `id`, `created_at`, `updated_at`
- `tenant_id` (FK Tenant — von welchem Tenant kam das Feedback)
- `tenant_user_id` (FK TenantUser — welcher User; nullable falls Feedback in nicht-eingeloggtem Kontext, in Phase 1 aber Pflicht da Widget nur eingeloggt funktioniert)
- `body` (Pflicht, Freitext-Beschreibung des Problems)
- `route_path` (Pflicht, der App-Pfad auf dem das Widget ausgelöst wurde, z.B. `/{tenant}/projects/abc-123`)
- `browser_info` (JSON: User-Agent, Viewport-Size, Screen-Size, Browser-Sprache, Timezone — automatisch befüllt)
- `app_version` (semver/Build-Hash der App-Version zum Zeitpunkt der Meldung)
- `attachments[]` (FK auf File-Records, siehe A.3)
- `status` (Enum: `open | in_progress | resolved | wontfix`)
- `assignee_admin_user_id` (FK auf AdminUser, nullable — Plattform-Admin der das Feedback bearbeitet)
- `internal_notes` (Markdown-Freitext, nur Plattform-Admin sichtbar)
- `resolved_at`, `resolved_by_admin_user_id`

Indizes:
- `(status, created_at DESC)` für Default-Liste

- `(tenant_id, created_at DESC)` für Filter pro Tenant
- `(assignee_admin_user_id, status)` für „meine offenen Tickets"

A.2 — Status-Lifecycle

```
open          → neu eingegangen, ungesehen oder gesehen aber unbearbeitet
in_progress   → Plattform-Admin hat sich zugewiesen, arbeitet dran
resolved      → behoben (Bug-Fix, Erklärung an User, etc.)
wontfix       → bewusst nicht behoben (Duplikat, kein Bug, falscher Kontext, etc.)
```

Übergänge:
- `open → in_progress`: Plattform-Admin ordnet sich zu
- `open/in_progress → resolved`: Plattform-Admin markiert als gelöst
- `open/in_progress → wontfix`: Plattform-Admin verwirft
- `resolved/wontfix → open`: Plattform-Admin re-opens

In Phase 1 keine Notifications zurück an den Tenant-User über Status-Änderungen.

A.3 — Attachments

- Wiederverwendet die `File`-Infrastruktur aus Modul 23 (Files / Dokumente)
- Storage-Pfad: `tenants/{tenantId}/feedback/{feedbackId}/...`
- Erlaubte Typen: PNG, JPG, WEBP, PDF (Phase 1)
- Max. 5 Attachments pro Feedback, max. 10 MB pro File
- Auto-Screenshot der aktuellen Page wird angeboten (Browser-API, optional, mit User-Bestätigung)

Block B — In-App-Feedback-Widget (Tenant-Seite)

B.1 — Auslöse-Funktion

Zu zeigende UI-Element-Anforderungen (Designer-Entscheidung wo platziert):
- Feedback-Trigger-Funktion („Problem melden"-Button/Icon) **überall in der App** verfügbar — von jeder Tenant-Route aus erreichbar (z.B. global in Top-Bar oder als Floating-Action-Button, Designer entscheidet)
- Gleiche Trigger-Funktion in `/help/`-Routen (öffentlich) — aber dort verlinkt auf mailto: `support@hivegrid.de` (kein Login → kein Tenant-Context)

B.2 — Eingabe-Modal

Zu zeigende Daten:
- Hinweis-Text *„Was hakt? Wir lesen jedes Feedback persönlich."*
- Aktuelle Route-Path-Anzeige (read-only, zeigt User wo er gerade ist — Transparenz, dass wir das mit-erfassen)
- Hinweis-Text dass Browser-Info + App-Version mitgeschickt werden (DSGVO-Transparenz)

Benötigte UI-Elemente:
- Multi-line-Text-Input für Beschreibung (Pflicht, min. 10 Zeichen)
- Attachment-Upload-Komponente (Drag&Drop + Klassischer-Datei-Wähler, max. 5 Files)
- „Screenshot der aktuellen Seite anhängen"-Funktion (One-Click, fügt automatisch generierten Screenshot als Attachment hinzu)
- Send-Button (disabled bis Beschreibung ≥ 10 Zeichen)
- Cancel-Button
- Klartext-Hinweis bei Send-Erfolg: *„Danke. Wir haben dein Feedback bekommen und melden uns falls nötig."*
- Klartext-Hinweis bei Fehler: *„Senden hat nicht geklappt — du kannst uns auch direkt schreiben: support@hivegrid.de"*

Operationen:
- Senden → POST an internen API-Endpoint → `Feedback`-Record erstellt → Plattform-Admin sieht in Liste
- Cancel → Modal schließt, kein Datenverlust-Warning (Phase 1 sehr basic)

B.3 — Auto-Befüllung (Code-internal)

Bei Submit wird automatisch erfasst:
- `tenant_id` aus aktuellem Tenant-Context
- `tenant_user_id` aus Session
- `route_path` aus aktuellem URL-Path (ohne Query-Params um keine Tokens/IDs durchzuleaken — Plattform-Admin entscheidet pro Fall ob Query-Param-Sample relevant ist; Phase 2 erweiterbar)
- `browser_info` aus User-Agent + Viewport (Standard-Browser-API)
- `app_version` aus Build-Hash (eingebettet zur Build-Zeit)
- `created_at` server-side

B.4 — Klartext-Hinweise

- Beschreibung zu kurz: *„Bitte mindestens 10 Zeichen — je mehr Details, desto schneller können wir helfen."*
- Attachment zu groß: *„Datei zu groß (max. 10 MB). Beschneide sie oder lass sie weg — wir können trotzdem helfen."*
- Senden fehlgeschlagen: *„Senden hat nicht geklappt — Internet-Verbindung prüfen oder direkt an support@hivegrid.de mailen."*
- Rate-Limit erreicht (max. 10 Submissions/User/h): *„Du hast schon einige Meldungen in kurzer Zeit gesendet. Bitte warte kurz oder schreib uns direkt."*

B.5 — Performance / Limit

- Rate-Limit pro `tenant_user_id`: max. 10 Submissions/h, hard-block sonst
- Body-Größenlimit: 10.000 Zeichen
- Submit ist async, Modal-Close direkt nach API-OK (keine Blockade im UI)

Block C — Support-Mailbox

C.1 — Adresse + Anbindung

- Inbound-Mailbox: `support@hivegrid.de`
- Routing: ankommende Mails landen in einer dedizierten Mail-Postbox (z.B. Google Workspace oder Scaleway-Mail-Service)
- Phase 1: Mailbox wird **außerhalb der App** (z.B. via Gmail-Web) gelesen — keine eigene Mail-UI in HiveGrid für Phase 1
- Plattform-Admin-Bereich (Modul 37) zeigt nur einen **Direkt-Link auf die externe Mailbox** + Klartext-Hinweis *„Externe Support-Mailbox via Google Workspace / [...]"*

Konsequenz: keine eigene Mail-Sync-Implementierung für Support-Mailbox in Phase 1 (Aufwand-Begrenzung).

C.2 — Verknüpfung Mail ↔ Feedback

In Phase 1 **keine** automatische Verknüpfung. Support-Email-Antworten sind manuell zu führen.

Phase 2: Mail-Sync via Email-Integration (Modul 9) — Mails landen direkt in `Feedback`-System mit Status-Tracking und Reply-Funktion.

Block D — Plattform-Admin-Feedback-Liste

D.1 — Routes

Sidebar-Hauptpunkt im Plattform-Admin (Modul 37): *Feedback / Support*.

```
admin.hivegrid.de/feedback/                  → Liste aller Feedback-Submissions
admin.hivegrid.de/feedback/:id               → Detail-View einer Submission
admin.hivegrid.de/feedback/mailbox           → Direkt-Link auf externe Support-Mailbox (Phase 1)
```

D.2 — Liste (`/feedback/`)

Zu zeigende Daten pro Eintrag:
- Eingangs-Datum/-Uhrzeit
- Tenant-Name
- User-Name + Email
- Erste 100 Zeichen des Body
- Anzahl Attachments
- Route-Path (verkürzt)
- Status (visuell unterscheidbar)
- Assignee (falls zugewiesen)

Benötigte UI-Elemente:
- Tabellarische Liste, sortierbar nach Datum (Default: neueste zuerst)
- Filter:
- Status (Multi-Select: open / in_progress / resolved / wontfix)
- Tenant (Auto-Complete aus Tenant-Liste)
- Assignee (Multi-Select)
- Zeitraum (Date-Range-Picker)
- Volltext-Suche über `body` + Tenant-Name + User-Name
- Pagination (50 pro Page, Cursor-basiert)
- Bulk-Auswahl + Bulk-Status-Wechsel + Bulk-Assign

Operationen:
- Klick auf Zeile → Detail-View
- Bulk-Status-Update
- Bulk-Assign zu Plattform-Admin
- Export als CSV (gefilterte Auswahl)

Performance:
- Default-Liste indexiert über `(status, created_at DESC)`
- Bei vielen offenen Tickets: separate „Meine offenen"-View mit Index `(assignee_admin_user_id, status, created_at DESC)`

D.3 — Detail-View (`/feedback/:id`)

Zu zeigende Daten:
- Vollständiger Body
- Attachments mit Vorschau (Bild-Inline-Anzeige, PDF-Download-Link)
- Tenant-Block: Name, Plan, Status, Direktlink ins Tenant-Admin (Modul 37)
- User-Block: Name, Email, Rolle im Tenant, Direktlink ins Tenant-Admin
- Route-Path mit klickbarem Direktlink (für Plattform-Admin der Impersonation hat — Modul 37)
- Browser-Info (User-Agent, Viewport, etc.) ausklappbar
- App-Version
- Status + Assignee (editierbar)

- Internal-Notes (Markdown-Freitext, nur Plattform-Admin)
- Activity-Log: alle Status-Wechsel, Assignee-Wechsel, Notes-Änderungen mit Timestamp + Actor

Benötigte UI-Elemente:
- Body-Anzeige (mit URL-Link-Detection, kein Markdown-Rendering — User-Eingaben sind plain text)
- Attachment-Preview-Komponente (Image/PDF)
- Inline-Edit für Status + Assignee + Internal-Notes
- Sektion „Tenant-Kontext" mit Direktlinks
- Sektion „Browser-/Tech-Kontext" (ausklappbar)
- Sektion „Aktivitäten" (Activity-Stream-Komponente analog Modul 1 Block E, scope: nur dieses Feedback)

Operationen:
- Status setzen (`open / in_progress / resolved / wontfix`)
- Assignee zuordnen (Plattform-Admin)
- Internal-Notes editieren (Markdown)
- Attachment downloaden
- Direktlink ins Tenant öffnen (Modul 37 Impersonation)
- Hard-Delete (mit Klartext-Warnung — nur in Ausnahmefällen, z.B. PII-Verstoß)

Klartext-Hinweise:
- Tenant inzwischen gelöscht: *„Dieser Tenant wurde inzwischen gelöscht — Daten zu User/Tenant nur noch eingeschränkt verfügbar."*
- User inzwischen entfernt: *„Dieser User ist nicht mehr Mitglied dieses Tenants."*

Block E — Permissions-Anwendung

E.1 — Tenant-Seite (Feedback senden)

Permission	Erlaubt
(keine spezielle Permission)	Jeder eingeloggte Tenant-User kann Feedback senden

Begründung: „Problem melden" ist eine basale Funktion, kein Premium-Feature, keine Berechtigung nötig. Rate-Limit greift technisch.

E.2 — Plattform-Admin-Seite (Feedback lesen + bearbeiten)

Permission	Erlaubt
`platform_admin.feedback.read`	Liste + Detail-View aller Feedbacks
`platform_admin.feedback.write`	Status setzen, Assignee zuordnen, Internal-Notes editieren
`platform_admin.feedback.delete`	Feedback hard-löschen

Bundle-Mapping:

Bundle	Atomare Permissions
`platform_admin.support.full`	`platform_admin.feedback.read` + `platform_admin.feedback.write`
`platform_admin.super`	alle

Anwendung in Routes:
- `admin.hivegrid.de/feedback/*` → `platform_admin.feedback.read` minimum

Klartext-Fehler:
- Plattform-Admin ohne Permission greift auf `/feedback/`: 404 (nicht 403)
- Tenant-User ohne Login greift auf Submit-API: 401 mit Klartext *„Bitte einloggen, um Feedback zu senden."*

Block F — Klartext-Hinweise / Fehler-Cases

- Submit-API-Fehler 5xx: *„Senden hat nicht geklappt — bitte direkt an support@hivegrid.de schreiben."*
- Rate-Limit-Hit (Tenant-User): *„Du hast schon einige Meldungen in kurzer Zeit gesendet. Bitte warte kurz oder schreib uns direkt."*
- Attachment-Upload-Fehler: *„Datei konnte nicht hochgeladen werden ({reason}). Probier ein anderes Format."*
- Tenant inaktiv (frozen/suspended): Submit blockiert mit Klartext *„Dein Workspace ist derzeit pausiert. Schreib uns direkt: support@hivegrid.de"*

Block G — Activity-Stream-Integration

Phase 1: **keine Tenant-seitige Activity** (Feedback ist privater Vorgang Tenant↔Plattform). Kein Eintrag im Tenant-Activity-Stream.

Plattform-Admin-Audit-Log (Modul 21 Plattform-Variante) loggt:
- `feedback.received` (System-Event bei Submit)
- `feedback.status_changed` (mit Diff)
- `feedback.assigned` (mit User-Diff)
- `feedback.deleted`

Block H — Phase 2 / spätere Features

- **In-App-Live-Chat mit Support** (Intercom-/Crisp-Style; entweder Eigen-Implementierung oder eingebettetes Drittprodukt)
- **Vollständiges Ticket-System** mit Status-Tracking + Notifications zurück an User (Email + In-App-Notification)
- **User-facing-Status-Page** im Tenant: Liste seiner eigenen Feedbacks mit Status, kann antworten
- **NPS-Surveys** (1x/Quartal, in-App-Pop-up, anonymisiert ans Plattform-Team)
- **CSAT-Tracking** pro Resolved-Feedback (1-5-Sterne nach Resolution)
- **Support-Mailbox-Sync in der App** über Modul 9 (Email-Integration) — Mails landen direkt in `Feedback`-Liste
- **Auto-Triage durch KI:** beim Submit klassifiziert KI das Feedback (Bug / Feature-Request / Frage / Beschwerde) und schlägt Zuordnung vor
- **Feedback-Voting:** ähnliche Feedbacks werden zusammengefasst, andere Tenants können „+1" geben (Public-Roadmap-Feel)
- **Integration mit Bug-Tracker** (Linear / Jira / GitHub-Issues): Feedback → Issue mit Backlink
- **Slack-Notification** an Plattform-Team-Channel bei neuem Feedback (Phase 1 evtl. schon manuell via Webhook anbindbar)

Block I — Architektur-Vorschriften (Querschnitt)

- **LP-6:** Anti-Agentur-Positionierung — niedrige Hürde Feedback zu senden ist Teil von „Selbst-Wartbarkeit". User soll spüren: wir hören zu.
- **LP-9:** Spec-Tiefe — Datenmodell + Permissions explizit festgelegt

- **LP-10:** Klartext-Diagnostik — alle Fehler-Cases mit menschenlesbarem Text
- **AV #6 (Audit-Trail-Default):** alle Status-/Assignee-/Notes-Änderungen geloggt
- **DSGVO-Transparenz:** User wird im Modal informiert was mitgeschickt wird (Browser-Info, App-Version, Route-Path)

Detail-Spec-Status: DRAFT

Phase-1-Modul bewusst klein. **Vor Bau:**
- Submit-API-Endpoint definieren (`POST /api/internal/feedback`)
- Auto-Screenshot-Implementierung wählen (z.B. `html2canvas` clientseitig, mit User-Consent)
- Verknüpfung mit `File` -Infrastruktur aus Modul 23 für Attachments
- Plattform-Admin-Permissions konkretisieren (im Rahmen Modul 37)

Offene Detail-Punkte (nicht-blockierend):
- Externe Support-Mailbox-Provider entscheiden (Google Workspace empfohlen, da konsistent mit Geerkens-Setup)
- Rate-Limit-Werte ggf. nach erstem Live-Feedback-Volumen anpassen
- Attachment-Storage-Pfad-Konvention bestätigen (sollte mit Modul 23 konsistent sein)
- Ob Submit auch ohne Login möglich sein soll (z.B. von `/help/` -Public-Routes) — Phase 1 Empfehlung: nein, Public verweist auf mailto-Link