```
module: 35
title: DSGVO-Export / Account-Löschung / Tenant-Daten-Portabilität
phase: 1
status: detail-spec-draft
last_updated: 2026-05-08
session_source: brainstorming-session-2026-05-07-2206.md
related_lps: [LP-8, LP-9, LP-10, LP-13]
spec_style: requirements-pur
```

Modul 35 — DSGVO-Export / Account-Löschung

> ***Detail-Spec gemäß LP-9.*** *Konsolidiert User-Account-Löschung (Apple-Pflicht seit 2022, bereits in Modul 27 Block D.3 spec'd), Tenant-Daten-Export (DSGVO-Auskunftsrecht + Vendor-Lock-in-Vermeidung), Tenant-Lösch-Workflow bei Kündigung, AVV-Vertrag im Onboarding, Subprozessoren-Liste und Privacy-Manifest.* ***LP-13 (EU-Stack komplett) ist hier Querschnitt.***

Themenblöcke

```
A. Datenmodell (GdprExportJob, TenantDeletionRequest, AvvAcceptance, Subprocessor)
B. Routes (Tenant-Admin-Scope und User-Scope)
C. User-Daten-Export (jeder User für sich)
D. Tenant-Daten-Export (Tenant-Admin)
E. User-Account-Löschung (Apple-Pflicht, Verweis auf Modul 27 D.3)
F. Tenant-Kündigung und Lösch-Workflow (30-Tage-Grace, Hard-Delete)
G. AVV-Vertrag im Onboarding
H. Subprozessoren-Liste
I. Privacy-Manifest (Capacitor-App)
J. Permissions
K. Klartext-Hinweise
L. Phase 2 Erweiterungen
```

Block A — Datenmodell

GdprExportJob

Tenant-scoped (RLS aktiv).

- `id`, `tenant_id`, `created_at`, `updated_at`
- `requested_by_user_id` (welcher User hat den Export angefordert)
- `scope` (Enum: `user_self | tenant_full`)
- `target_user_id` (bei `scope=user_self`: der User dessen Daten gemeint sind; gleich `requested_by_user_id`)
- `status` (Enum: `queued | processing | ready | failed | expired`)
- `progress_percent` (Integer 0-100)
- `result_file_id` (FK auf `File` aus Modul 23, ZIP-Output)
- `result_size_bytes`
- `result_expires_at` (Default 7 Tage nach `ready` — danach automatischer Hard-Delete der ZIP-Datei)
- `download_count` (für Audit)
- `last_downloaded_at`
- `error_message` (Klartext bei `failed`)

- `processing_started_at`, `processing_finished_at`

Indizes: `(tenant_id, status)`, `(result_expires_at)` für Cleanup-Cron, `(requested_by_user_id, scope)`.

Constraint: max. 1 aktiver `tenant_full`-Job pro Tenant gleichzeitig. Max. 3 aktive `user_self`-Jobs pro User pro 24h (Rate-Limit gegen Missbrauch).

TenantDeletionRequest

Tenant-scoped (RLS aktiv).

- `id`, `tenant_id`, `created_at`
- `requested_by_user_id` (Tenant-Admin)
- `reason` (Enum: `voluntary | non_payment | suspended_violation | admin_initiated`) — `voluntary` ist Self-Service, Rest kommt aus Plattform-Admin
- `reason_text` (optionaler Freitext-Kommentar)
- `grace_period_ends_at` (Default `created_at + 30 Tage`)
- `status` (Enum: `pending | cancelled | executing | completed | failed`)
- `cancelled_by_user_id`, `cancelled_at` (bei Rücknahme innerhalb Grace-Period)
- `execution_started_at`, `execution_finished_at`
- `failure_reason` (bei `failed`)

Indizes: `(tenant_id)` (unique über `status IN ('pending', 'executing')` — max 1 aktive Lösch-Anforderung), `(grace_period_ends_at, status)` für Cron.

AvvAcceptance

Tenant-scoped, einmalig pro Tenant beim Onboarding angelegt.

- `id`, `tenant_id`, `created_at`
- `accepted_by_user_id` (User der die Checkbox geklickt hat — typischerweise GF beim Signup)
- `accepted_ip`, `accepted_user_agent` (für Beweisbarkeit)
- `avv_version` (z.B. `2026-05`) — bei AVV-Updates wird neuer Eintrag verlangt
- `pdf_file_id` (FK auf `File`, signiertes PDF mit Tenant-Daten + Datum)
- `signer_full_name`, `signer_company_role` (z.B. *„Geschäftsführer"*) — für PDF-Branding

Indizes: `(tenant_id, avv_version)` unique.

Subprocessor

Plattform-globale Tabelle (kein RLS, App-Code-gegated).

- `id`, `created_at`, `updated_at`
- `name` (z.B. *„Scaleway S.A.S."*)
- `purpose_text` (Klartext: *„Hosting + Object-Storage + DB"*)
- `country` (ISO, z.B. `FR`)
- `is_eu_eea` (Boolean)
- `data_categories[]` (z.B. *„Stamm-/Vertragsdaten, User-Content, Logs"*)
- `dpa_url` (Link auf öffentliches Auftragsverarbeitungs-Vertragstemplate des Subprozessors)
- `active_since` (Datum)
- `archived_at` (wenn aufgegeben)

Beim Onboarding: Tenant sieht aktuelle Liste als Teil der AVV-Akzeptanz.

Klartext-Sichtbarkeit von Drittland-Transfers

- `country` außerhalb EU/EEA → in Tenant-UI **Klartext-Warnung** sichtbar (*„Daten werden außerhalb der EU verarbeitet — Rechtsgrundlage: ..."*)

- In Phase 1 ist das EU-Stack-Pflicht (LP-13), daher Liste sollte ausschließlich EU/EEA-Anbieter enthalten — Schema unterstützt aber Drittland für Sichtbarkeit/Korrektheit

---

Block B — Routes

Tenant-Admin-Scope (Workspace-Verwaltung)

```
/{tenant}/admin/privacy/                        → Privacy-Hub (Übersicht)
/{tenant}/admin/privacy/export/                 → Tenant-Daten-Export-Liste
/{tenant}/admin/privacy/export/new              → Neuen Export anstoßen
/{tenant}/admin/privacy/export/:jobId           → Export-Job-Status
/{tenant}/admin/privacy/deletion/               → Tenant-Lösch-Workflow starten
/{tenant}/admin/privacy/deletion/:requestId     → Lösch-Anforderungs-Status (mit Cancel-Funktion)
/{tenant}/admin/privacy/subprocessors/          → Subprozessoren-Liste
/{tenant}/admin/privacy/avv/                    → AVV-PDF-Download und Akzeptanz-History
```

Permission: `tenant.admin` für alle.

User-Scope (persönliche Settings)

```
/{tenant}/settings/account/                     → Account-Settings (siehe Modul 27 D.3)
/{tenant}/settings/account/data-export          → Eigener User-Daten-Export
/{tenant}/settings/account/data-export/:jobId   → Export-Job-Status
/{tenant}/settings/account/delete               → Account-Löschung (Multi-Step, siehe Modul 27
D.3)
```

Permission: Login + Tenant-Membership (jeder User darf seine eigenen Daten exportieren und seinen Account löschen).

Block C — User-Daten-Export (User-Scope)

Anwendungsfall

Jeder User darf jederzeit alle Daten, die mit seiner Person verknüpft sind, in maschinenlesbarem Format anfordern (DSGVO Art. 15 Auskunftsrecht + Art. 20 Datenportabilität).

Was wird exportiert

Stammdaten:
- User-Account-Stammdaten (Email, Name, Sprache, 2FA-Status, OAuth-Verknüpfungen, Notification-Preferences)
- Aktive Sessions (gekürzt: nur Geräte-Liste, keine Tokens)
- API-Keys (gekürzt: keine Secret-Werte)

Tenant-Membership-Daten:
- TenantUser-Beziehung (Rollen, Groups, Beitritts-Datum)
- Permission-Snapshot

User-erzeugte Inhalte:
- Activities, in denen User als Actor auftritt (`actor_user_id = self`)
- Notes, die User erstellt hat
- Chat-Nachrichten, die User geschrieben hat
- Tasks, die User erstellt oder zugewiesen wurden
- Termine (Booking + Calendar) mit User-Beteiligung

- Hochgeladene Files (Metadaten + Binary)
- Form-Submissions (falls von User stammend)
- HR-Datenbestand zu sich selbst (eigene Krankmeldungen, Urlaubsanträge, Payslips, Personaldokumente — nicht die der Kollegen)

Kommunikations-Daten:
- Outbound-Emails, die User gesendet hat (mit Body)
- Eingehende Emails, die in Mailbox des Users waren

KI/Audit:
- Eigene KI-Konversations-Historie (HiveMind, AI-Assistent)
- Audit-Spuren mit User als `actor_user_id`

Format

ZIP-Archiv mit Struktur:

```
user-export-{userId}-{tenantSlug}-{YYYY-MM-DD}.zip
├── README.md                    ← Klartext-Erklärung was im ZIP ist
├── manifest.json                ← Schema-Version, Generations-Datum, Tenant, User
├── account/
|   ├── profile.json
|   ├── sessions.json
|   ├── notification-preferences.json
|   └── oauth-connections.json
├── memberships/
|   └── tenant-{tenantId}.json
├── activities/
|   └── activities.jsonl         ← JSONL für große Mengen
├── chat/
|   └── messages.jsonl
├── notes/
|   └── notes.jsonl
├── tasks/
|   └── tasks.jsonl
├── appointments/
|   └── appointments.jsonl
├── files/
|   ├── files-metadata.jsonl
|   └── binaries/                ← Original-Files in Subordnern pro Modul
|       ├── projects/
|       ├── chat/
|       └── ...
├── emails/
|   ├── outbound.jsonl
|   ├── inbound.jsonl
|   └── attachments/             ← Original-Attachments
├── hr/
|   ├── leave-requests.jsonl
|   ├── sick-reports.jsonl
|   ├── payslips/                ← Original-PDFs
|   └── personal-docs/
├── ai-conversations/
|   └── conversations.jsonl
└── audit/
    └── audit-trail.jsonl
```

JSON-Schema-Version im `manifest.json` für künftige Kompatibilität.

Workflow

1. User klickt *„Daten exportieren"* in `/{tenant}/settings/account/data-export`
2. Klartext-Erklärung: was wird exportiert, ungefähre Dauer, Aufbewahrungs-Frist
3. Bestätigungs-Klick → `GdprExportJob` mit `scope=user_self` wird angelegt, Status `queued`
4. Worker (BullMQ) verarbeitet: zieht Daten aus allen Modulen, packt ZIP, lädt in Storage hoch
5. Status `ready` → User bekommt Email mit Download-Link (Magic-Link gegen Auth-Bypass) + In-App-Notification
6. Download via signed-URL (max. Lebenszeit 7 Tage, dann ZIP gelöscht)

Edge-Cases

- User hat sehr große Datenmenge (z.B. 50.000+ Activities): Worker arbeitet in Chunks, Progress-Update alle 5%
- Job läuft länger als 24h: Status auf `failed` setzen, Klartext: *„Export hat länger als erwartet gedauert. Bitte erneut anstoßen oder Support kontaktieren."*
- Concurrent: max. 3 aktive `user_self`-Jobs pro User pro 24h (Block mit Klartext)
- Permission-Check: User darf nur seinen eigenen Export anfordern. Tenant-Admin darf **nicht** für andere User-Daten exportieren (das wäre `tenant_full`-Scope)

Block D — Tenant-Daten-Export (Tenant-Admin-Scope)

Anwendungsfall

Tenant-Admin (`tenant.admin`) darf alle Daten des kompletten Tenants in maschinenlesbarem Format anfordern. Zwei Hauptzwecke:
- DSGVO-Auskunftsrecht für die Firma als Verantwortlicher
- Vendor-Lock-in-Vermeidung (Wechsel weg von HiveGrid)

Was wird exportiert

Alle tenant-scoped Entitäten als JSON. Für jede Entität ein eigenes JSONL-File:

- `crm/contacts.jsonl` , `crm/companies.jsonl` , `crm/leads.jsonl` , `crm/contact-company-relations.jsonl`
- `crm/custom-field-definitions.jsonl` , `crm/custom-field-values.jsonl`
- `deals/deals.jsonl` , `deals/pipelines.jsonl` , `deals/pipeline-stages.jsonl`
- `projects/projects.jsonl` , `projects/project-pipelines.jsonl` , `projects/project-members.jsonl` , `projects/acceptance-protocols.jsonl` , `projects/schedule-entries.jsonl`
- `tasks/tasks.jsonl`
- `forms/forms.jsonl` , `forms/form-submissions.jsonl` , `forms/form-actions.jsonl`
- `bookings/booking-pages.jsonl` , `bookings/appointments.jsonl` , `bookings/availability-rules.jsonl`
- `email/inbound.jsonl` , `email/outbound.jsonl` , `email/campaigns.jsonl` , `email/sending-domains.jsonl`
- `chat/messages.jsonl` , `chat/threads.jsonl` , `chat/reactions.jsonl`
- `whatsapp/...` (falls aktiv)
- `telephony/call-sessions.jsonl`
- `signatures/...`
- `hr/...` (alle HR-Hub-Daten)
- `portal/...` (Customer-Portal)
- `automations/rules.jsonl` , `automations/executions.jsonl`
- `ai/conversations.jsonl` , `ai/embeddings-metadata.jsonl` (Embeddings selbst nicht exportiert — nicht-portabel)
- `kwp/...` (KWP-Sync-Daten falls aktiv)

- `audit/audit-trail.jsonl`
- `users/tenant-users.jsonl` (mit gekürzten PII anderer User — Email, Name; nicht Passwort, nicht 2FA-Secrets)
- `roles-permissions/roles.jsonl`, `roles-permissions/permissions.jsonl`, `roles-permissions/groups.jsonl`
- `tags/tags.jsonl`, `tags/entity-tags.jsonl`
- `feature-flags/effective-values.jsonl` (welche Flags waren wann aktiv)
- `subscriptions/subscription-history.jsonl`

Plus alle Files im Original-Format unter `files/` (gleiche Verzeichnis-Struktur wie im Storage: `files/{module}/{entityId}/{fileName}`).

Format

ZIP-Archiv (potenziell sehr groß — kann mehrere GB sein):

```
tenant-export-{tenantSlug}-{YYYY-MM-DD}.zip
├── README.md
├── manifest.json
├── schema/
|   └── *.json                    ← JSON-Schema-Definitionen je Entität
├── crm/
├── deals/
├── projects/
├── ... (siehe oben)
└── files/
```

Bei extrem großen Tenants: Multi-Part-ZIP mit Index-File (`tenant-export-...part01.zip`, `part02.zip`, ...). Implementierungs-Detail.

Workflow

1. Tenant-Admin klickt *„Tenant-Daten exportieren"* in `/{tenant}/admin/privacy/export/new`
2. Klartext-Erklärung: was wird exportiert, geschätzte Größe (basierend auf DB-/Storage-Statistik), erwartete Dauer, Aufbewahrungs-Frist (7 Tage Download-Window)
3. Optional: Scope-Filter (z.B. *„nur CRM"*, *„nur HR"*) — **Phase 2**, in Phase 1 immer Vollexport
4. Bestätigungs-Klick → `GdprExportJob` mit `scope=tenant_full`
5. Worker arbeitet asynchron — kann Stunden dauern bei großen Tenants
6. Progress sichtbar in Real-Time (Websocket oder Polling)
7. Status `ready` → Tenant-Admin bekommt Email + In-App-Notification
8. Download via signed-URL, Multi-Part-Download wenn nötig
9. Audit-Eintrag mit `download_count` pro Zugriff

Edge-Cases

- Tenant ist im Status `frozen` oder `past_due`: Export trotzdem erlaubt (Lese-Recht hat Vorrang)
- Tenant ist `deletion_pending`: Export erlaubt während Grace-Period — Klartext-Hinweis *„Letzte Möglichkeit, Daten zu sichern bevor Tenant gelöscht wird."*
- Job-Fehler mitten im Export: Resume-Mode (Worker speichert Checkpoint, kann ab letzter erfolgreicher Entität fortsetzen)
- Storage-Limit überschritten beim Schreiben des ZIPs: Klartext-Fehler *„Export-ZIP zu groß für Storage. Bitte Support kontaktieren — wir splitten es in Teilexporte."*
- Concurrent: max. 1 aktiver `tenant_full`-Job pro Tenant
- Klartext bei extrem großem Tenant beim Anstoßen: *„Geschätzte Dauer: ca. 4-6 Stunden. Geschätzte Größe: ca. 12 GB. [Trotzdem starten]"*

Block E — User-Account-Löschung

Spec ist in Modul 27 Block D.3 vollständig. Hier nur Konsolidierungs-Hinweise:

- Multi-Step-Bestätigung in `/{tenant}/settings/account/delete`
- Apple-Pflicht seit 2022 — App-Store-Compliance gegeben durch Sichtbarkeit auf gleicher Ebene wie Profil-Settings
- Was passiert technisch: TenantUser-Beziehungen gelöscht, PII mit `[deleted-{userId}]` überschrieben, Audit-Spuren behalten User-ID, 2FA/Sessions/Push-Tokens/OAuth-Konten weg, Avatar gelöscht
- 24h-Reverse-Window mit Bestätigungs-Email an Lösch-Adresse

Erweiterung in Modul 35:
- Bei User-Löschung wird **automatisch** ein User-Daten-Export angeboten (Multi-Step Step 1.5: *„Möchtest du vorher deine Daten exportieren? [Ja, Export starten] [Nein, direkt löschen]"*)
- User klickt „Ja" → Export-Job läuft, User bekommt Email mit Download → erst nach Download (oder nach Ablauf 7 Tage) wird Lösch-Workflow fortgesetzt
- User klickt „Nein" → direkter Übergang zu Step 2 (Identitäts-Bestätigung)

Block F — Tenant-Kündigung und Lösch-Workflow

Voluntary Deletion (Self-Service durch Tenant-Admin)

Workflow:

1. Tenant-Admin in `/{tenant}/admin/privacy/deletion/` klickt *„Workspace kündigen"*
2. **Step 1 — Erklärung:** Klartext was passiert
   - Status wechselt sofort auf `deletion_pending`
   - Workspace bleibt 30 Tage sichtbar (Read-Only-Modus, Login möglich, Daten lesbar, Schreib-Aktionen blockiert)
   - Nach 30 Tagen Hard-Delete aller Tenant-Daten inkl. Files
   - Ab Hard-Delete: nicht wiederherstellbar
3. **Step 2 — Datensicherung-Hinweis:** *„Vor Lösch-Bestätigung empfehlen wir: Tenant-Daten-Export starten. [Export jetzt starten] [Habe ich bereits] [Daten brauche ich nicht]"*
4. **Step 3 — Stripe-Hinweis:** Klartext über Restbetrag, Pro-rata-Erstattung, letzte Rechnung
5. **Step 4 — Identitäts-Bestätigung:** Pwd + 2FA falls aktiv
6. **Step 5 — Final-Klick** mit Tenant-Slug-Eingabe als Confirm-Phrase (User muss `geerkens-gmbh` eintippen, nicht nur einen Knopf drücken)
7. `TenantDeletionRequest` mit `reason=voluntary`, `grace_period_ends_at = now + 30 Tage` wird angelegt
8. Tenant-Status → `deletion_pending`
9. Bestätigungs-Email an alle Tenant-Admins
10. Banner im Workspace: *„Konto wird in X Tagen gelöscht — [Kündigung zurückziehen]"*

Cancel innerhalb Grace-Period

- Jeder Tenant-Admin kann mit einem Klick rückgängig machen → Status zurück auf `active` (oder vorherigen Status)
- Audit-Eintrag: wer hat gekündigt, wer hat zurückgezogen
- Stripe-Subscription wird bei Cancel reaktiviert (siehe Modul 28)

Hard-Delete-Cron (nach 30 Tagen)

Cron-Job läuft täglich, prüft `TenantDeletionRequest` mit `grace_period_ends_at < now AND status='pending'`.

Für jeden Hit:

1. Status auf `executing`, `execution_started_at` gesetzt

2. **Reihenfolge der Löschung** (FK-Abhängigkeit beachten):
   - Files in Scaleway-Storage: `tenants/{tenantId}/...` rekursiv löschen
   - Embeddings (pgvector): alle `TenantEmbedding`-Einträge löschen
   - Worker-Queues: alle ausstehenden Jobs für diesen Tenant verwerfen
   - Webhook-Subscriptions: deaktivieren und löschen
   - DB-Daten: Reihenfolge per FK — Activity-Daten → Verknüpfungs-Tabellen → Haupt-Entitäten → User-Memberships → Tenant selbst
   - Audit-Daten: behalten in Plattform-Audit-Log (Modul 37) ohne Tenant-Daten — nur Metadata (Lösch-Datum, User-ID des Auslösers, Reason)
   - Cache-Keys: Redis `tenant:{tenantId}:*` purgen
3. Status auf `completed`
4. Tenant-Status → `deleted` (siehe Modul 27 A.5)
5. Bestätigungs-Email an `accepted_by_user_id` aus AVV: *„Tenant ‚X' wurde am [Datum] vollständig gelöscht. DSGVO-Auftrag erfüllt."*
6. Plattform-Audit-Log-Eintrag

Was bleibt nach Hard-Delete

- Audit-Metadata in `AdminAuditLog` (Modul 37): Tenant-ID, Slug, Lösch-Datum, Reason, Akteur
- Stripe-Records (von Stripe-Seite, gemäß deren Aufbewahrungs-Pflichten)
- Email-Logs vom Email-Provider (gemäß deren Retention)
- **Keine** PII, **keine** User-Inhalte

Reasons außer `voluntary`

- `non_payment` — automatisch nach 30 Tagen `past_due` (Trigger: Stripe-Webhook-Logik in Modul 28); `TenantDeletionRequest` wird automatisch angelegt
- `suspended_violation` — Plattform-Admin kündigt manuell wegen AGB-Verstoß; ggf. verkürzte Grace-Period
- `admin_initiated` — Plattform-Admin kündigt im Auftrag des Tenant-Admins (z.B. Email-Anfrage); Tenant-Admin bekommt Bestätigungs-Email mit Cancel-Link

Edge-Cases

- Hard-Delete-Cron schlägt mittendrin fehl: Status `failed`, Resume-fähig (idempotente Lösch-Aktionen)
- Tenant-Daten-Export läuft noch beim Hard-Delete: Export-Job wird abgebrochen, ZIP gelöscht, Klartext im Audit
- Tenant-Admin kündigt während aktiver Subscription: Stripe-Cancel wird beim Hard-Delete-Cron getriggert (nicht beim Anlegen der `TenantDeletionRequest` — User soll noch zurückziehen können ohne Stripe-Doppel-Abbuchung)
- Last-Tenant-Admin kündigt Tenant: erlaubt (anders als bei User-Löschung — Tenant-Kündigung ist explizit das Ende)

Block G — AVV-Vertrag im Onboarding

Akzeptanz beim Signup

In Modul 27 A.1 Self-Service-Signup-Formular: Checkbox „AVV-Akzeptanz" als Pflicht-Feld.

Erweiterung hier:

- Beim Klick: PDF-Vorschau des AVV mit aktueller Version (`avv_version`)
- PDF wird **dynamisch generiert** mit Tenant-spezifischen Daten:
- Firma (Tenant-Name)
- Anschrift (aus Signup)
- GF-Name + Rolle (Signer)

- Datum
- Aktuelle Subprozessoren-Liste (Snapshot der Tabelle)
- Aktuelle Version-ID
- User klickt *„Elektronisch unterschreiben"* → System speichert:
- `AvvAcceptance`-Eintrag mit IP, User-Agent, Datum, Signer-Name, Version
- Generiertes PDF in Storage (`tenants/{tenantId}/legal/avv-{version}-{date}.pdf`)
- Audit-Eintrag

Re-Akzeptanz bei Updates

- Bei AVV-Update (neue `avv_version`): Plattform-Admin triggert Re-Akzeptanz für alle Tenants
- Beim nächsten Login: Modal mit Klartext *„Unsere AVV wurde aktualisiert (Änderungen: [Klartext]). Bitte erneut akzeptieren."*
- Nutzungs-Sperre nach 14 Tagen ohne Re-Akzeptanz → Read-Only-Modus mit Klartext-Banner
- Tenant-Admin sieht Liste aller akzeptierten Versionen im `/admin/privacy/avv/`

Download

- Tenant-Admin sieht in `/{tenant}/admin/privacy/avv/`:
- Aktuelle Version (PDF-Download)
- History aller akzeptierten Versionen mit Datum + Signer
- Download als PDF jederzeit möglich

Edge-Cases

- AVV-Update während aktiver Trial-Phase: Tenant muss akzeptieren bevor Trial in `active` übergeht
- Tenant lehnt Re-Akzeptanz ab: Klartext mit Folgen (*„Ohne Akzeptanz können wir den Service nicht weiter erbringen — bitte Support kontaktieren oder Workspace kündigen"*) und Link zu Lösch-Workflow

---

Block H — Subprozessoren-Liste

Sichtbarkeit für Tenant

`/{tenant}/admin/privacy/subprocessors/` zeigt:

- Liste aller aktiven `Subprocessor`-Einträge
- Pro Eintrag: Name, Land, EU/EEA-Indikator, Zweck-Klartext, Datenkategorien, DPA-Link, „Aktiv seit"

Benötigte UI-Elemente:
- Tabellarische Anzeige
- Filter: nur EU vs. alle, nach Datenkategorie
- Such-Funktion über Name + Zweck
- Export als CSV/PDF (für Tenant-eigene Compliance-Doku)

Änderungs-Benachrichtigung

- Bei Hinzufügen eines neuen Subprozessors: Plattform-Admin triggert Benachrichtigung
- Email an alle Tenant-Admins mit Klartext-Beschreibung des neuen Subprozessors
- 14-Tage-Widerspruchs-Frist (gesetzliche AVV-Pflicht): Tenant kann widersprechen → Klartext mit Folgen (*„Ohne neuen Subprozessor können wir Feature X nicht anbieten — bitte Support kontaktieren"*)
- Audit-Eintrag pro Notification-Versand

Plattform-Admin-Verwaltung

In Modul 37: Subprocessoren-Anlegen / Editieren / Archivieren mit Cross-Tenant-Auswirkung. Detail dort.

Block I — Privacy-Manifest (Capacitor-App)

Apple-Pflicht seit 2024. Modul-übergreifender Querschnitt — wird in Modul 36 (Mobile-App) konkret implementiert, aber Inhalt-Quelle liegt hier:

Pflicht-Inhalte

- **Daten-Sammelarten:** welche Datentypen sammelt die App (Kontakt-Info, User-Content, Identifiers, Diagnostics, Usage-Data)
- **Tracking-Domains:** Liste aller Drittanbieter-Domains, an die Daten gehen
- **Required-Reason-APIs:** für jeden genutzten Apple-API mit Reason-Pflicht (z.B. `UserDefaults`, `FileTimestamp`) der dokumentierte Grund

Dynamisch vs. statisch

- **Statisch:** generelle App-Funktion (Kamera, Push, Location-bei-Nutzung)
- **Dynamisch:** je nach aktivierten Modulen pro Tenant — z.B. Aircall-Telefonie aktiv → Telefon-Permissions; HR-Hub aktiv → Personal-Daten; etc.
- In Phase 1 wird ein **maximales Privacy-Manifest** verwendet (alle möglichen Datenarten deklariert), unabhängig davon ob Tenant das Feature aktiv hat. Spätere Optimierung Phase 2.

Lebenszyklus

- Bei jedem App-Store-Submit muss Privacy-Manifest aktuell sein
- Plattform-Admin pflegt Master-Liste in Modul 37
- Build-Pipeline für Capacitor-App generiert `PrivacyInfo.xcprivacy` aus Master-Liste

Klartext-Sichtbarkeit für Tenant

- `/{tenant}/admin/privacy/` zeigt Link auf öffentliche Privacy-Manifest-Seite (Marketing-Site `hivegrid.de/privacy-manifest`)
- Quick-Verlinkung auf Drittland-Übersicht (siehe Block H)

Block J — Permissions

Permission-IDs

Permission	Erlaubt
`privacy.user_export.self`	Eigenen User-Daten-Export anstoßen (jeder eingeloggte Tenant-User implizit)
`privacy.tenant_export`	Tenant-Daten-Export anstoßen
`privacy.tenant_deletion`	Tenant-Lösch-Workflow starten oder zurückziehen
`privacy.subprocessors.read`	Subprozessoren-Liste einsehen
`privacy.avv.read`	AVV-PDF und Akzeptanz-History einsehen
`privacy.avv.accept`	AVV (re-)akzeptieren

Bundle-Mapping

Bundle	Atomare Permissions
`privacy.user` (jeder eingeloggte User implizit)	`privacy.user_export.self`
`privacy.tenant_admin` (kommt mit `tenant.admin`)	alle obigen außer `user_export.self`-Beschränkung

Anwendung in Routes

- `/{tenant}/settings/account/data-export` → eingeloggter User
- `/{tenant}/settings/account/delete` → eingeloggter User
- `/{tenant}/admin/privacy/export/*` → `privacy.tenant_export`
- `/{tenant}/admin/privacy/deletion/*` → `privacy.tenant_deletion`
- `/{tenant}/admin/privacy/subprocessors/` → `privacy.subprocessors.read`
- `/{tenant}/admin/privacy/avv/` → `privacy.avv.read`

Sonderfall: User-Daten-Export für andere User

- **Phase 1:** nicht möglich. Jeder User exportiert nur seine eigenen Daten.
- **Phase 2:** Tenant-Admin kann auf Anfrage des Users (z.B. Mitarbeiter ist krank, braucht Daten) im Vertretungsfall anstoßen — mit zusätzlicher Bestätigungs-Email an betroffenen User und Audit-Eintrag.

Block K — Klartext-Hinweise / -Fehler

- Export-Job läuft länger als 24h: *„Dein Export dauert länger als erwartet. Wir versuchen es weiter — du bekommst eine Email sobald fertig. Bei Problemen: [Support kontaktieren]"*
- Export-ZIP abgelaufen (>7 Tage): *„Dieser Export ist abgelaufen und wurde gelöscht. [Neuen Export starten]"*
- Concurrent-Limit: *„Du hast bereits 3 aktive Export-Anforderungen in den letzten 24 Stunden. Bitte warte bis einer fertig ist."*
- Tenant-Lösch-Workflow ohne aktive Subscription: *„Dein Tenant ist im Trial — Kündigung beendet sofort den Zugang ohne Grace-Period. [Trotzdem fortfahren]"*
- Hard-Delete-Cron läuft: Banner *„Tenant-Daten werden gerade endgültig gelöscht. Bitte nicht schließen. Geschätzte Dauer: X Min."* (für Plattform-Admin sichtbar)
- AVV-Re-Akzeptanz blockiert: *„Bitte unsere aktualisierte AVV akzeptieren, um den Workspace weiter zu nutzen. [Anschauen + Akzeptieren] [Workspace kündigen]"*
- Subprozessor-Widerspruchs-Frist abgelaufen: *„Du hast nicht widersprochen. Der neue Subprozessor wird ab [Datum] wirksam."*
- Privacy-Manifest-Update bei App-Update: *„Diese App-Version verarbeitet zusätzliche Datenarten: [Liste]. Mehr erfahren: [Link]"* (Apple-Required-Disclosure)

Block L — Phase 2 / spätere Features

- **Pseudonymisierungs-Toggle** — Tenant kann pro Modul wählen, dass exportierte Daten pseudonymisiert werden (z.B. Email-Adressen gehasht)
- **Drittland-Audit-Log** — wenn Subprozessor außerhalb EU/EEA, separater Log mit Begründung pro Datentransfer
- **DSFA-Generator** (Datenschutz-Folgeabschätzung) — automatisch generiertes PDF basierend auf aktivierten Modulen + Subprozessoren + Datenkategorien für Tenant-Compliance
- **Scope-gefilterter Export** — nur CRM, nur HR, etc. (statt immer Vollexport)
- **Inkrementelle Exports** — *„nur Daten seit letztem Export"* für regelmäßige Sicherungen

- **Auto-Delete-Schedules** — DSGVO-Speicherbegrenzung: z.B. *„lösche Leads im Status* `lost` *automatisch nach 90 Tagen"* — pro Entity-Typ konfigurierbar
- **User-Daten-Export für andere User durch Tenant-Admin** (siehe Block J)
- **Verschlüsselter Export** — User legt Passwort fest, ZIP wird AES-verschlüsselt
- **Export-API** — programmatischer Zugriff via API-Key (für regelmäßige automatische Backups durch Tenant)
- **Konsolidierte Privacy-Center-Page** — Tenant sieht alle DSGVO-relevanten Aspekte in einer Übersicht (Aufbewahrungs-Fristen, aktive Datenflüsse, offene Lösch-Anforderungen)

Block M — Architektur-Vorschriften (Querschnitt)

- **LP-13 — EU-Stack komplett:** alle Subprozessoren EU/EEA in Phase 1
- **LP-10 — Klartext-Diagnostik:** alle Lösch-/Export-Hinweise menschenlesbar (Block K)
- **AV #6 — Audit-Trail-Default:** alle Privacy-Aktionen geloggt (Export-Anstoß, Download, Lösch-Anforderung, AVV-Akzeptanz, Subprozessor-Notification)
- **LP-8 — App-Store-Strategie:** User-Account-Löschung in App (Apple-Pflicht 2022), Privacy-Manifest (Apple-Pflicht 2024)
- **LP-9 — Spec-Tiefe:** keine UI-Layouts

Detail-Spec-Status: DRAFT

Erste vollständige Detail-Spec für Modul 35. Konsolidiert User-Account-Löschung aus Modul 27 D.3 und ergänzt Tenant-Daten-Export, Tenant-Lösch-Workflow, AVV-Vertrag, Subprozessoren-Liste und Privacy-Manifest.

Offene Detail-Punkte (nicht-blockierend):
- AVV-PDF-Template (rechtlicher Inhalt) finalisieren mit Anwalt — Schema steht
- Multi-Part-ZIP-Strategie für sehr große Tenants (Implementierungs-Detail)
- Privacy-Manifest-Master-Liste konkrete Apple-API-Reasons (Implementierungs-Detail in Modul 36)
- Konkrete Subprozessoren-Liste zum Launch (Plattform-Admin-Pflege)
- Storage-Pfad-Konvention für Lösch-Audit-Daten in Plattform-Audit-Log final mit Modul 37 abstimmen
- Subframe-UI-Designs separat