```
module: 3
title: Sales-Pipeline (Deals)
phase: 1
status: detail-spec-draft
last_updated: 2026-05-08
session_source: brainstorming-session-2026-05-07-2206.md
related_lps: [LP-3, LP-4, LP-7, LP-9, LP-10]
spec_style: requirements-pur
```

Modul 3 — Sales-Pipeline (Deals)

> ***Detail-Spec gemäß LP-9.*** *Kern-Vertriebs-Modul. Löst HubSpot-Pipeline-Tool ab (LP-7). Baut auf M1 (CRM-Kern) und M2 (Lead-Inbox) auf, übergibt an M4 (Auftragspipeline) bei Won-Handoff.*

Themenblöcke

```
A. Datenmodell (Pipeline, Stages, Deal, DealActivity, HiveNumberSeq)
B. Menü-Struktur und Routes
C. Pipeline-Verwaltung (Multi-Pipeline-Konfiguration, Stages)
D. Kanban-View / Listen-View
E. Deal-Detail-View
F. Deal-Lifecycle (Anlegen, Stage-Wechsel, Won/Lost, Archivierung)
G. Forecast & KPIs
H. Integration mit anderen Modulen
I. Permissions-Anwendung (mit Field-Level-Permissions)
```

Block A — Datenmodell

Architektur-Entscheidung: Pipeline ist Entität, Multi-Pipeline ab Tag 1

Pipeline wird als eigene Entität gespeichert, nicht als reiner View/Filter. Tenant kann mehrere Pipelines mit unterschiedlichen Stages anlegen. Begründung: Verkaufsfälle (Anlagen-Verkauf vs. Wartungsverträge vs. Reparaturen) haben unterschiedliche Stages — und Geerkens kommt von HubSpot, kennt das Pattern.

Kern-Entitäten

```
Pipeline                — eine Sales-Pipeline pro Tenant (mehrere möglich)
PipelineStage           — Phase innerhalb einer Pipeline
Deal                    — ein Verkaufsfall in einer Pipeline
DealForm                — Verknüpfung Deal ↔ Form-Submission
LostReason              — Tenant-konfigurierbare Lost-Gründe
HiveNumberSeq           — fortlaufende Deal-Nummern pro Tenant
```

Pipeline

- `id`, `tenant_id`, `created_at`, `updated_at`
- `name` (z.B. „Anlagen-Verkauf", „Wartungsverträge")
- `description` (optional)
- `is_default` (Boolean — eine pro Tenant ist Default für neue Deals ohne Pipeline-Auswahl)

- `position` (Sortierung im UI)
- `archived_at`

PipelineStage

- `id`, `tenant_id`, `pipeline_id`
- `name` (z.B. „Anfrage", „Besichtigung", „Angebot", „Verhandlung", „Gewonnen", „Verloren")
- `position` (innerhalb Pipeline)
- `stage_kind` (Enum: `open | won | lost`) — Markiert Stage-Typ; `won` / `lost` sind Terminal-Stages
- `win_probability` (0-100 % — für Forecast; bei `won` automatisch 100, `lost` automatisch 0)
- `auto_archive_after_days` (optional — Stuck-Detection oder Auto-Archivierung)
- `color` (visueller Hinweis; Designer entscheidet konkrete Wahl)
- `archived_at`

Validation: Pipeline muss mind. 1 `won`- und 1 `lost`-Stage haben (UI-Validation beim Speichern).

Deal

- `id`, `tenant_id`, `created_at`, `updated_at`, `created_by_user_id`
- `deal_number` (fortlaufend über `HiveNumberSeq`, z.B. „D-2026-0042")
- `pipeline_id`, `stage_id`
- `title` (Pflicht)
- `description` (Markdown-Freitext)
- `volume` (`numeric(12,2)` — Cent-Genauigkeit, UI zeigt Default ganze Euro)
- `currency` (Default EUR)
- `expected_close_date` (geschätzter Abschluss-Termin)
- `actual_close_date` (gesetzt bei Won/Lost)
- `assigned_to_user_id` (Verantwortlicher Vertriebler)
- `contact_id` (FK auf primären Kontakt — Pflicht)
- `company_id` (FK auf primäre Firma, optional bei B2C)
- `lead_id` (FK auf ursprünglichen Lead, falls aus Promotion)
- `lost_reason_id` (FK auf LostReason, nur bei Stage = `lost`, **Pflicht** in dem Fall)
- `won_handoff_project_id` (FK auf Projekt, gesetzt nach Won-Handoff zu M4)
- `archived_at`

DealActivity (Schreibt in zentrale `Activity`-Tabelle aus M1)

Activity-Typen pro Deal:
- `deal_created`
- `deal_stage_changed` (Metadata: `from_stage_id`, `to_stage_id`, `duration_in_previous_stage_seconds`)
- `deal_volume_changed`
- `deal_assignee_changed`
- `deal_pipeline_changed`
- `deal_won` / `deal_lost`
- `note_added`
- Alle CRM-Activities mit verknüpftem Contact erscheinen automatisch im Deal-Stream

DealForm

- `deal_id`, `form_submission_id`, `linked_at`
- Verknüpft Deal mit Form-Submissions

LostReason (Tenant-konfigurierbar)

- `id`, `tenant_id`, `name`, `archived_at`
- Tenant-Admin verwaltet in `/admin/sales/lost-reasons/`
- SHK-Template seedet 5 Standard-Reasons (z.B. „Zu teuer", „Konkurrenz", „Kein Bedarf mehr", „Kein Budget", „Nicht erreicht")

HiveNumberSeq

- `id`, `tenant_id`, `entity_type` (`deal | project | invoice | ...`)
- `current_number` (fortlaufend)
- `prefix` (z.B. „D-2026-")
- `format` (z.B. `{prefix}{number:04d}` → „D-2026-0042")
- Reset-Strategie: Phase 1 nie reset; Phase 2 jährlich konfigurierbar

SHK-Template-Pipelines (beim Tenant-Anlegen geseedet)

Pipeline 1: „Anlagen-Verkauf"
1. Anfrage (open, 10 %)
2. Besichtigung (open, 30 %)
3. Bedarfsanalyse (open, 50 %)
4. Angebot (open, 70 %)
5. Verhandlung (open, 85 %)
6. Gewonnen (won, 100 %)
7. Verloren (lost, 0 %)

Pipeline 2: „Wartungsverträge"
1. Erstkontakt (open, 30 %)
2. Vertrag-Versand (open, 60 %)
3. Unterschrift erhalten (open, 90 %)
4. Aktiv (won, 100 %)
5. Abgesagt (lost, 0 %)

Reparatur-Aufträge laufen direkt als Auftrag (M4), nicht über Sales-Pipeline.

Block B — Menü-Struktur und Routes

Sidebar-Hauptpunkt: *Sales* (oder direkt *Deals* — Designer-Entscheidung)

```
/{tenant}/deals                          → Default-Pipeline (Kanban der is_default=true Pipeline)
/{tenant}/deals/list                     → Alle Deals als Tabelle (Cross-Pipeline-Sicht)
/{tenant}/deals/pipelines                → Pipeline-Switcher / Übersicht
/{tenant}/deals/pipelines/:id            → Spezifische Pipeline-Ansicht (Kanban)
/{tenant}/deals/pipelines/:id/list       → Spezifische Pipeline als Tabelle
/{tenant}/deals/new                      → Neuer Deal (mit Pipeline-Vorauswahl wenn aus Pipeline-
View)
/{tenant}/deals/:id                      → Deal-Detail
/{tenant}/deals/archive                  → Archivierte Deals (Won/Lost + soft-archived)

/{tenant}/admin/sales/pipelines          → Pipeline-Verwaltung
/{tenant}/admin/sales/pipelines/:id      → Einzelne Pipeline editieren (Stages, Defaults)
/{tenant}/admin/sales/lost-reasons       → Lost-Reasons-Liste
```

Default-Routen-Logik:
- `/{tenant}/deals` öffnet immer die `is_default=true`-Pipeline
- Bei Multi-Pipeline-Tenants: Pipeline-Switcher-Komponente im Header der Sales-View

---

Block C — Pipeline-Verwaltung

Pipeline-Liste (`/{tenant}/admin/sales/pipelines/`)

Zu zeigende Daten:
- Liste aller Pipelines pro Tenant
- Pro Pipeline: Name, Anzahl Stages, Anzahl aktiver Deals, is_default-Markierung, Status

Benötigte UI-Elemente:
- Listen-Anzeige
- Drag-&-Drop für Pipeline-Reihenfolge
- Funktion: neue Pipeline anlegen
- Funktion: Pipeline duplizieren
- Funktion: Pipeline editieren / archivieren / als Default markieren

Operationen:
- Pipeline anlegen (Name, Description, optional aus Template)
- Pipeline duplizieren (alle Stages werden mitkopiert, neue IDs)
- Pipeline archivieren (Block falls aktive Deals — Reassign-Workflow)
- Pipeline löschen (hard, nur wenn keine Deals jemals drin waren)
- Pipeline als Default markieren

Pipeline-Editor (`/{tenant}/admin/sales/pipelines/:id`)

Zu zeigende Daten:
- Pipeline-Stammdaten (Name, Description, is_default)
- Liste der Stages mit allen Feldern

Benötigte UI-Elemente:
- Editor-Funktion für Stammdaten
- Stage-Liste mit Drag-&-Drop für Reihenfolge
- Add-Funktion für neue Stage
- Edit-/Delete-Funktion pro Stage
- Validation-Anzeige für Won-/Lost-Stage-Pflicht

Operationen:
- Stage hinzufügen (Name, Stage-Kind, Win-Probability, Color, auto_archive_after_days)
- Stage editieren
- Stage löschen (Block falls noch Deals — Reassign-Workflow nötig)
- Stages umsortieren
- Pipeline als Default markieren

Klartext-Validation:
- *„Diese Pipeline hat keine ‚Gewonnen'-Stage. Bitte mindestens eine Stage vom Typ ‚won' anlegen."*
- *„Diese Pipeline hat keine ‚Verloren'-Stage. Bitte mindestens eine Stage vom Typ ‚lost' anlegen."*
- Stage löschen mit Deals: *„In dieser Stage sind 8 aktive Deals. Bitte zuerst neu zuweisen oder Stage archivieren statt löschen."*

Lost-Reasons-Verwaltung (`/{tenant}/admin/sales/lost-reasons/`)

Zu zeigende Daten: Liste mit Name, Anzahl verknüpfter Deals, Status.

Benötigte UI-Elemente: Standard-Listen-Editor mit Add/Edit/Archive.

Operationen: Standard-CRUD; Archivieren statt Hard-Delete (Reporting-Konsistenz).

Block D — Kanban-View / Listen-View

Kanban-View (`/{tenant}/deals/pipelines/:id` oder Default `/{tenant}/deals`)

Zu zeigende Daten:
- Spalten entsprechen Stages der gewählten Pipeline (in `position`-Reihenfolge)
- Pro Stage-Spalte: Stage-Name, Anzahl Deals, Aggregat-Volumen, Win-Probability
- Pro Deal-Karte:
- Deal-Titel
- Volumen (gegated über `deals.read_volume_*` — siehe Block I)
- Verantwortlicher (Avatar/Initialen)
- Erwarteter Close-Date
- Tags (max. 3 sichtbar, Rest als „+N"-Hinweis)
- Visueller Hinweis bei überzogenem Close-Date
- Visueller Hinweis bei Stage-Stuck

Benötigte UI-Elemente:
- Spalten-Layout (Designer entscheidet horizontal scrollbar / responsive)
- Drag-&-Drop-Funktion zwischen Spalten
- Filter-Leiste mit Multi-Select für: Verantwortlicher, Tags, Volumen-Range, Erwarteter Close-Date Range
- Such-Funktion (Volltext über Titel + Description + verknüpfte Kontakt-/Firmen-Namen)
- Pipeline-Switcher
- Toggle „Nur meine Deals" (Quick-Filter)
- Klick auf Deal-Karte öffnet Deal-Detail
- Bulk-Auswahl für Mass-Actions
- Pagination pro Stage-Spalte: bei 100+ Deals „Mehr laden"-Funktion

Operationen — Drag-&-Drop:
- Drag von Spalte A nach Spalte B → `deal_stage_changed`-Activity (Metadata: `from_stage_id`, `to_stage_id`, `duration_in_previous_stage_seconds`)
- Bei Drag in `won`-Stage: Bestätigungs-Dialog mit Won-Handoff-Option (Default ja, opt-out möglich)
- Bei Drag in `lost`-Stage: Bestätigungs-Dialog erzwingt `lost_reason`-Auswahl
- Reverse-Window: Won/Lost-Status kann innerhalb 24h zurückgenommen werden

Operationen — Bulk:
- Tags hinzufügen/entfernen
- Verantwortlichen ändern
- Stage-Wechsel (Bulk-Move)
- Archivieren

Listen-View (`/{tenant}/deals/list` oder `/{tenant}/deals/pipelines/:id/list`)

Zu zeigende Daten pro Eintrag:
- Deal-Nummer, Titel, Pipeline (bei Cross-Pipeline-Sicht), Stage, Verantwortlicher, Volumen (gegated), Verknüpfter Kontakt + Firma, Tags, Erwarteter Close-Date, Letzte Activity

Benötigte UI-Elemente:
- Tabellarische Anzeige
- Spalten-Header für Sortierung
- Filter wie in Kanban
- Spalten-Konfiguration pro User (gespeichert)

- Bulk-Auswahl
- Export als CSV (gefilterte Auswahl)
- Pagination (Cursor-basiert, 50/Page, „Mehr laden")

Default-Sortierung: zuletzt geändert absteigend.

Cross-Pipeline-Listen-View (`/{tenant}/deals/list`)

Gleiche Funktionalität, zeigt Deals aus allen Pipelines mit zusätzlicher Pipeline-Spalte und Pipeline-Filter.

Performance-Anforderungen

- Server-side-Filter (kein Client-side bei 1.000+ Deals)
- Index auf `tenant_id, pipeline_id, stage_id, archived_at, last_activity_at`
- Aggregat-Cache pro Stage (Anzahl + Volumen-Summe), invalidiert bei Stage-Wechsel
- Real-Time-Updates: bei Stage-Wechsel durch andere User wird Kanban Live aktualisiert (WebSocket oder Server-Sent-Events)

Block E — Deal-Detail-View

Deal-Detail (`/{tenant}/deals/:id`)

Zu zeigende Daten:
- Deal-Stammdaten (Deal-Nummer, Titel, Description, Volumen, Pipeline, Stage, erwarteter Close-Date, tatsächlicher Close-Date)
- Verknüpfter Kontakt + Firma
- Verantwortlicher
- Lead-Origin-Hinweis (falls aus Promotion mit Link zu Lead)
- Tags
- Custom-Felder (separate Definitionen mit `applicable_to=deal`)
- Stage-Historie (Liste aller Stage-Wechsel mit Datum und Dauer)
- Activity-Stream (M1 Block E)
- Verknüpfte Files
- Verknüpfte Form-Submissions
- Verknüpfte Tasks
- Verknüpftes Projekt (falls Won-Handoff erfolgt — Link zu M4)

Benötigte UI-Elemente:
- Sektionierte Anzeige der Stammdaten
- Inline-Edit für editierbare Felder
- Stage-Anzeige als visuelle Pipeline-Komponente (zeigt aktuelle Position, Klick auf andere Stage triggert Stage-Wechsel)
- Activity-Stream-Komponente
- Tab- oder Sektions-Navigation für: Übersicht / Activities / Files / Tasks / Forms / Custom-Felder
- Action-Funktionen für die unten gelisteten Operationen
- Markdown-Editor für Description und Notizen

Operationen:
- Stammdaten editieren (Inline)
- Stage wechseln (mit Bestätigungs-Workflow für Won/Lost)
- Verantwortlichen ändern
- Pipeline wechseln (Edge-Case mit Klartext-Hinweis über Historie)
- Verknüpften Kontakt/Firma ändern
- Tags hinzufügen/entfernen
- Custom-Feld-Werte setzen

- Click-to-Call (über AirCall-Integration)
- Email schreiben (Compose mit Empfänger vorausgefüllt)
- Termin buchen
- Aufgabe anlegen
- Notiz hinzufügen
- File hochladen
- Form-Submission verknüpfen
- Archivieren / Reaktivieren
- Hard-Delete

Won-Handoff-Workflow:
- Bei Stage-Wechsel zu `won`: Bestätigungs-Dialog *„Auftrag/Projekt aus diesem Deal anlegen?"*
- Bei Bestätigung:
- Trigger M4 zur Projekt-Anlage
- Mapping: Deal-Felder → Projekt-Felder (Titel, Volumen, Verantwortlicher, Kontakt, Firma, verknüpfte Files, Notizen)
- Wahl der Ziel-Auftrags-Pipeline + Start-Stage
- `Deal.won_handoff_project_id` wird gesetzt
- Bei Ablehnung: Deal nur als Won markiert, kein Projekt
- Reverse innerhalb 24h: Won-Status zurück + ggf. Projekt löschen

Lost-Workflow:
- Bei Stage-Wechsel zu `lost`: Bestätigungs-Dialog erzwingt `lost_reason`-Auswahl
- Optional: Freitext-Notiz
- Reverse innerhalb 24h möglich

Klartext-Hinweise:
- Won-Stage: Banner *„Dieser Deal wurde als gewonnen markiert am [Datum]. [Won-Status zurücknehmen — 24h verfügbar]"*
- Lost-Stage: Banner *„Dieser Deal ist als verloren markiert. Grund: [Reason]. [Wieder öffnen]"*
- Stuck: Banner *„Dieser Deal ist seit X Tagen in Stage [Name] — typisch sind Y Tage. Soll eine Erinnerung erstellt werden?"*
- Verknüpfter Kontakt archiviert: Hinweis *„Der zugeordnete Kontakt ist archiviert"* mit Reaktivierungs-Link

---

Block F — Deal-Lifecycle

Deal-Anlage-Quellen

Manuelle Anlage (`/{tenant}/deals/new`):
- Pflichtfelder: Titel, Pipeline (Default = is_default), Verantwortlicher (Default = current_user), Verknüpfter Kontakt (Auto-Complete-Suche aus M1)
- Optionale Felder: Volumen, erwarteter Close-Date, Description, Tags, verknüpfte Firma, Custom-Felder
- Bei Anlage: Deal startet in der ersten Stage (`position=0`) der gewählten Pipeline
- Deal-Nummer wird automatisch generiert

Aus Lead-Promotion (M1):
- Promotion-Modal optional „Deal anlegen" (Default ja)
- Lead-Daten werden auf Deal-Felder gemappt
- `Deal.lead_id` zeigt auf Lead, Lead bekommt `converted_to_deal_id`

Aus Form-Submission (M5):
- Form-Action „Deal anlegen" konfigurierbar pro Form
- Field-Mapping vom Tenant-Admin definiert
- Submission wird über `DealForm` verknüpft

Aus Email-Conversation (M9):
- „Deal aus dieser Email anlegen"-Funktion in Email-Detail
- Auto-Match: existierender Kontakt mit Email → vorausgewählt
- Email-Subject wird Deal-Titel-Vorschlag

Aus Anruf (M12):
- AirCall-Webhook nach Anruf → optionale Deal-Anlage in Anruf-Detail
- Auto-Match Telefon → Kontakt
- Anruf-Notiz wird Deal-Description-Vorschlag

Aus Automation (M17):
- Action `create-deal` mit konfigurierbarem Mapping
- Trigger z.B. „neuer Lead aus Quelle X"

Stage-Wechsel-Validation

- Vorwärts-Wechsel: erlaubt
- Rückwärts-Wechsel: erlaubt mit Klartext-Hinweis
- Wechsel zu Won/Lost: Bestätigungs-Workflow (siehe Block E)
- Wechsel von Won/Lost zurück: nur 24h Reverse-Window
- Pipeline-Wechsel: erfordert `tenant.admin` oder `deals.admin`, Activity wird mit Diff geschrieben

Auto-Archive

- Won-Deals: nach 30 Tagen ins Archiv (konfigurierbar pro Pipeline)
- Lost-Deals: nach 90 Tagen ins Archiv (konfigurierbar pro Pipeline)
- Archivierte Deals bleiben in Reports/KPIs sichtbar
- Tenant-Admin kann Auto-Archive deaktivieren

Stage-Stuck-Detection (Cron)

- `auto_archive_after_days` pro Stage konfigurierbar
- Daily-Cron markiert Deals, die länger als Schwelle in einer Stage sind
- Stuck-Status zeigt sich als visueller Hinweis im Kanban + Banner im Detail
- Optional: Automation-Trigger `deal.stuck.in_stage`

Reaktivierung archivierter Deals

- Tenant-Admin oder Verantwortlicher kann reaktivieren
- Reaktivierung setzt `archived_at = null`, führt Deal in ursprüngliche oder gewählte Stage zurück
- Activity-Eintrag

Hard-Delete

- Erfordert `deals.admin`
- Multi-Step-Confirm mit Klartext-Warnung über Anzahl verknüpfter Tasks/Files/Activities
- Falls Deal mit Won-Handoff verknüpft (`won_handoff_project_id != null`): Block, *„Dieser Deal hat ein verknüpftes Projekt. Bitte zuerst Verknüpfung lösen oder Projekt selbst archivieren/löschen."*
- Audit-Trail bleibt nach Hard-Delete (Hash-Eintrag mit Lösch-Datum + User)

Block G — Forecast & KPIs

> *Diese Metriken werden im Dashboard-Modul (M20) als Widgets gezeigt. Sales-Pipeline stellt Datenquellen und Berechnungen bereit.*

Forecast-Berechnung

Pipeline-Forecast (pro Pipeline und cross-pipeline):
- Summe über alle aktiven Deals (nicht in Won/Lost-Stage, nicht archiviert): `volume × stage.win_probability / 100`
- Beispiel: Deal über 25.000 EUR in Stage „Verhandlung" mit `win_probability=70` → trägt 17.500 EUR zum Forecast bei
- Aufteilbar nach Zeitraum (anhand `expected_close_date`): „dieser Monat", „dieses Quartal", „nächste 12 Monate"

Realisierter Umsatz:
- Summe `volume` aller Deals mit `stage_kind=won` und `actual_close_date` im Zeitraum
- Vergleichbar zum Forecast als Genauigkeits-Indikator

KPIs pro Vertriebler

- Aktive Deals (Anzahl + Gesamt-Volumen)
- Forecast-Beitrag (gewichtetes Volumen)
- Won-Rate (`won / (won + lost)` der letzten 90 Tage)
- Aktivitäts-Metriken (basierend auf Activity-Stream): Anzahl Anrufe, Mails, Termine
- Durchschnittliche Cycle-Time (Tage von Anlage bis Won)
- Stuck-Deals (Anzahl in Stuck-Status)

Stage-Conversion-Rates

Pro Pipeline: *„Wie viele % der Deals, die je in Stage A waren, kommen in Stage B an?"* — berechnet aus `stage_changed`-Activities.

Lost-Reason-Auswertung

Anteilige Verteilung der Lost-Reasons.

Performance-Anforderungen

- Aggregat-Cache pro Stage (Deal-Count + Volumen-Summe)
- Stage-Verweildauer-Index auf Activity-Tabelle
- Forecast-Cache pro Pipeline + Zeitraum
- KPI-Berechnung server-side

Block H — Integration mit anderen Modulen

Lead-Promotion (M1)

- Lead → Deal-Anlage über Promotions-Modal in M1
- `Deal.lead_id` gesetzt, Lead-Activity-Stream wandert in Deal-Activity-Stream

Won-Handoff zu Auftragspipeline (M4)

- Bei Stage-Wechsel zu `won` mit Handoff-Bestätigung: Projekt automatisch in M4 angelegt
- Mapping: Deal-Felder → Projekt-Felder

- `Deal.won_handoff_project_id` gesetzt

Form-Submission → Deal (M5)

- Form-Action „Deal anlegen" pro Form konfigurierbar
- Field-Mapping vom Tenant-Admin definiert
- Submission über `DealForm` verknüpft

Email-Sync → Deal (M9)

- Eingehende Mail mit Bezug zu verknüpftem Kontakt erscheint im Activity-Stream
- Manuelle „Deal aus Mail anlegen"-Funktion
- Email-Subject-Pattern (z.B. `[D-2026-0042]`) → Auto-Verknüpfung

Telefonie → Deal (M12)

- AirCall-Anrufe werden mit Kontakt verknüpft, erscheinen im Activity-Stream
- Click-to-Call aus Deal-Detail
- Anruf-Notiz nach Anruf editierbar

Automation-Engine (M17)

Trigger aus Sales-Pipeline:
- `deal.created`
- `deal.stage_changed` (mit `from_stage`, `to_stage`)
- `deal.won` / `deal.lost`
- `deal.stuck.in_stage` (Cron-basiert)
- `deal.assignee_changed`

Actions für Sales-Pipeline:
- `create-deal`
- `update-deal-stage`
- `update-deal-volume`
- `assign-deal-to-user`
- `create-project-from-deal` (Won-Handoff)

Tasks (M26)

- Tasks können mit Deal verknüpft werden
- Sichtbar im Deal-Detail Tab „Tasks"

Custom-Felder (M1 Block F)

- Deal hat eigene Custom-Definitionen mit `applicable_to=deal`
- SHK-Template-Defaults: *„Anlagen-Typ"* (select), *„Wärmebedarf kW"* (number), *„Bestandsanlage"* (boolean)

Block I — Permissions-Anwendung (mit Field-Level-Permissions)

Object-Level-Permission-IDs

Permission	Erlaubt
`deals.read_own`	Deals sehen, bei denen User = `assigned_to_user_id`
`deals.read_all`	Alle Deals sehen
`deals.write`	Anlegen + Bearbeiten (Permissions-aware)
`deals.change_stage`	Stage-Wechsel (kann separat sein, z.B. Junior darf editieren aber nicht Won markieren)
`deals.delete`	Hard-Delete
`deals.admin`	Pipeline-Verwaltung, Lost-Reasons-Verwaltung, Bulk-Operations, Forecast-Reports

Field-Level-Permission-IDs

Permission	Erlaubt
`deals.read_volume_own`	Volumen *eigener* Deals sehen
`deals.read_volume_all`	Volumen *aller* sichtbaren Deals sehen
`deals.read_aggregates`	Aggregierte KPIs (Forecast, Stage-Volumen, Sales-Performance-Charts) sehen
`deals.read_close_date_all`	Erwartete Close-Dates aller Deals sehen (sonst nur eigene)

Bundle-Mapping (Variante 2)

Bundle	Atomare Permissions
`deals.none`	(keine)
`deals.read_own`	`deals.read_own` + `deals.read_volume_own`
`deals.read_all`	`deals.read_own` + `deals.read_all` + `deals.read_volume_own`
`deals.write`	`deals.read_all` + `deals.read_volume_all` + `deals.read_aggregates` + `deals.read_close_date_all` + `deals.write` + `deals.change_stage`
`deals.admin`	alle obigen + `deals.delete` + `deals.admin`

Field-Level-Permissions können im Advanced-Mode atomar überschrieben werden für Spezialfälle.

SHK-Template-Default-Rollen

Rolle	Deal-Permissions
GF	`deals.admin` (alles inkl. Volumen + Aggregate)
Vertriebsleiter	`deals.write` + `read_volume_all` + `read_aggregates`
Vertriebler	`deals.read_all` + `read_volume_own` + `write_own` (sieht alle Deals, aber Volumen nur bei eigenen)
Projektleiter / Bauleiter	`deals.read_all` (sieht Deals als Kontext, **kein** Volumen, **keine** Aggregate)
Büro	`deals.read_all` + `read_volume_all` (für Rechnungen/Aufträge)
Marketing	`deals.read_all` + `read_aggregates` (Conversion-Analyse, **ohne** Volumen-Detail)
Monteur	`deals.none`

UI-Anwendung der Field-Level-Permissions

- Volumen-Feld ohne Read-Permission: zeigt „—" oder ist ausgeblendet (Kanban-Karte + Detail-View + Listen-Spalte)
- Aggregat-Volumen pro Stage-Spalte im Kanban: ausgeblendet ohne `read_aggregates`
- Forecast-Widget im Dashboard: ausgeblendet ohne `read_aggregates`
- Custom-Felder mit sensitiven Daten (z.B. Marge): kommen in Phase 2 mit Field-Level-Permission pro Custom-Feld

Anwendung in Routes

- `/{tenant}/deals` → `deals.read_own` oder `deals.read_all`
- `/{tenant}/deals/list` → analog
- `/{tenant}/deals/pipelines/:id` → analog
- `/{tenant}/deals/new` → `deals.write`
- `/{tenant}/deals/:id` → `deals.read_own` (mit Owner-Check) oder `deals.read_all`
- `/{tenant}/deals/archive` → analog
- `/{tenant}/admin/sales/pipelines` → `tenant.admin` ODER `deals.admin`
- `/{tenant}/admin/sales/lost-reasons` → analog

Klartext-Fehler

- Zugriff auf nicht-eigenen Deal mit `read_own`: 404 (nicht 403)
- Stage-Wechsel ohne `change_stage`: Server-side abgelehnt, Klartext: *„Du darfst Deals nicht in andere Phasen verschieben. Bitte einen Vertriebsleiter."*
- Won-Handoff ohne `projects.write`: Klartext: *„Du kannst diesen Deal als gewonnen markieren, aber das automatische Anlegen eines Projekts erfordert zusätzliche Berechtigung. Möchtest du nur den Won-Status setzen?"*
- Volumen-Feld ohne Read-Permission: kein expliziter Fehler, einfach „—" angezeigt (Permissions-aware-Default, AV #5)

Architektur-Vorschriften (Querschnitt)

- **LP-3, AV #3:** SHK-Pipelines + Lost-Reasons sind Branchen-Template-Seeding, nicht Hardcode
- **LP-4:** Sales-Pipeline ist Commodity (HubSpot-Ablösung) — sauber bauen, aber kein Über-Engineering
- **LP-10, AV #1:** Klartext-Diagnostik überall (Stage-Validation, Stuck-Hinweise, Won/Lost-Bestätigung)
- **AV #5:** Permissions-aware-Default in allen Listen/Filtern
- **AV #6:** Audit-Trail-Default für alle schreibenden Aktionen (insb. Stage-Wechsel)

Detail-Spec-Status: DRAFT

Erste vollständige Detail-Spec für Modul 3. **Vor Bau:** Verifizierung mit Noah. Bei Implementierung dann konkrete Index-Definitionen, Migrations-SQL und Caching-Strategie ergänzen.

Offene Detail-Punkte:
- Konkrete Index-Strategie pro Tabelle
- WebSocket vs. SSE für Real-Time-Kanban-Updates (Boilerplate-Entscheidung)
- Cache-Invalidierungs-Strategie für Aggregate (Pub/Sub vs. Trigger-basiert)
- Konkrete Won-/Lost-Reverse-Mechanik (Soft-Revert vs. neue Stage-Activity)
- Konkrete SHK-Stage-Probabilities mit Noah finalisieren