```
module: 26
title: Tasks / Aufgaben
phase: 1
status: detail-spec-draft
last_updated: 2026-05-08
session_source: brainstorming-session-2026-05-07-2206.md
related_lps: [LP-2, LP-4, LP-7, LP-9, LP-10]
spec_style: requirements-pur
```

Modul 26 — Tasks / Aufgaben

> ***Detail-Spec gemäß LP-9.*** *Leichtgewichtiges Aufgaben-Modul, das quer zu allen Entitäten liegt. Tasks entstehen manuell, aus Chat-Nachrichten („Mark as Task"), aus Form-Submissions und aus Automation-Actions. Commodity-Säule (LP-4) — gut genug, kein Wettbewerb mit Asana/Trello.*

Themenblöcke

```
A. Datenmodell
B. Menü-Struktur und Routes
C. Listen-Views (Meine Tasks, Projekt-Tasks, Tenant-Tasks)
D. Detail-View
E. Erstellungs-Quellen (manuell, Chat, Form, Automation)
F. Reminder & Auto-Archivierung
G. Activity-Stream-Integration
H. Permissions-Anwendung
```

Block A — Datenmodell

Task

- `id`, `tenant_id`, `created_at`, `updated_at`, `created_by_user_id`
- `title` (Pflicht, Text)
- `description` (Markdown-Freitext, optional)
- `assigned_to_user_id` (FK auf User, optional — eine Person als Hauptverantwortlicher)
- `assigned_to_group_id` (FK auf Group, optional — alternativ zu Person; siehe Constraint)
- `deadline_at` (Timestamp, optional)
- `priority` (Enum: `low | normal | high | urgent`, Default `normal`)
- `status` (Enum: `open | in_progress | done`, Default `open`)
- `completed_at` (Timestamp, gesetzt bei Übergang zu `done`)
- `completed_by_user_id` (FK auf User, gesetzt bei Übergang zu `done`)
- `archived_at` (Timestamp, gesetzt bei Auto-Archivierung)
- `source` (Enum: `manual | chat_message | form_submission | automation`)
- `source_ref` (JSON — Referenz auf Quell-Entität, z.B. `{ "chat_message_id": "..." }`)
- `linked_entity_type` (Enum: `contact | company | lead | deal | project | null`)
- `linked_entity_id` (FK, optional — die Haupt-Entität, an der die Task hängt)

Constraints:
- Entweder `assigned_to_user_id` oder `assigned_to_group_id` gesetzt — nicht beides, nicht keines (Tasks ohne

Zuweisung sind nicht erlaubt; bei Anlage Default = Ersteller)
- `completed_at` und `completed_by_user_id` nur gesetzt wenn `status = done`
- Bei Wechsel `done → open/in_progress`: `completed_at` und `completed_by_user_id` werden zurückgesetzt

Indizes:
- `(tenant_id, assigned_to_user_id, status, deadline_at)` — für „Meine Tasks"-Listen
- `(tenant_id, linked_entity_type, linked_entity_id)` — für Anzeige an Entity-Detail-Views
- `(tenant_id, status, completed_at)` — für Auto-Archivierungs-Cron
- `(tenant_id, archived_at)` — für Filter „Archivierte anzeigen"

TaskWatcher (Phase 1, optional erweiterbar)

- `task_id`, `user_id`, `tenant_id`
- M:N — User, die Notifications zur Task bekommen sollen, ohne Assignee zu sein
- Auto-Eintrag: Ersteller wird Watcher, wenn nicht selbst Assignee
- Phase 1 minimal genutzt — Erweiterung in Phase 2

Block B — Menü-Struktur und Routes

Sidebar-Hauptpunkt: *Aufgaben* (eigener Top-Level-Punkt, da quer zu allen Modulen).

Routen:

```
/{tenant}/tasks                  → Task-Liste (Default: meine offenen Tasks)
/{tenant}/tasks/new              → Neue Task (Modal oder Page – Designer entscheidet)
/{tenant}/tasks/:id              → Task-Detail (Modal oder Page – Designer entscheidet)
```

Sub-Routen / Filter-Presets (über Query-Parameter, keine eigenen Routen):
- `?scope=mine` (Default) — vom User zugewiesene Tasks
- `?scope=created_by_me` — vom User erstellte Tasks
- `?scope=all` — alle Tasks im Tenant (nur mit `tasks.read_all`)
- `?status=open|in_progress|done|archived`
- `?linked_entity_type=project&linked_entity_id=...` — Tasks zu einer bestimmten Entität (verlinkt aus Entity-Detail-Views)

Embed-Anzeige in Entity-Detail-Views:
- Contact-/Company-/Lead-/Deal-/Project-Detail-Views zeigen einen Tasks-Tab/Section mit verknüpften Tasks
- Funktion „Aufgabe anlegen" in jeder Entity-Detail-View (Pre-Fill der Verknüpfung)

Block C — Listen-Views

„Meine Tasks" (Default-Ansicht unter `/{tenant}/tasks`)

Zu zeigende Daten pro Eintrag:
- Titel
- Priorität (visuelles Zeichen — Designer entscheidet)
- Deadline (relativ wenn ≤ 7 Tage, absolut sonst, Markierung wenn überfällig)
- Verknüpfte Entität (Typ + Name als Klick-Link)
- Assignee (Person oder Group — bei Group-Zuweisung als Group-Name)
- Status
- Quelle (visueller Hinweis, falls aus Chat/Form/Automation entstanden)

Benötigte UI-Elemente:
- Listen- oder Karten-Anzeige (Designer entscheidet)
- Quick-Actions pro Eintrag: Status auf `done` setzen, Priorität ändern, Deadline ändern
- Such-Funktion (Volltext über Titel + Beschreibung)
- Filter-Komponenten:
- Status (Multi-Select)
- Priorität (Multi-Select)
- Deadline (Date-Range, Presets: heute / diese Woche / überfällig / kein Datum)
- Verknüpfungs-Typ (Multi-Select: Contact/Company/Lead/Deal/Project/keine)
- Quelle (Multi-Select)
- Toggle „Archivierte anzeigen"
- Sortierung: Default Deadline aufsteigend, sekundär Priorität, alternativ nach Erstellt-Datum
- Bulk-Auswahl (Checkboxes pro Zeile + Master-Checkbox)
- Pagination (Cursor-basiert, „Mehr laden")
- View-Switcher zwischen Scope-Presets (mine / created_by_me / all wenn berechtigt)

Operationen:
- Status-Wechsel inline
- Bulk-Actions: Status ändern, Priorität ändern, Assignee ändern, Archivieren, Löschen
- Klick auf Eintrag → Detail-View

Performance:
- Server-side-Filter
- Default-Limit 50 pro Page

„Projekt-Tasks" / „Deal-Tasks" / „Contact-Tasks" (Embed in Entity-Detail-Views)

Zu zeigende Daten: wie oben, aber Verknüpfungs-Spalte entfällt (impliziert).

Operationen:
- Anlage einer Task mit Pre-Fill der Verknüpfung
- Quick-Actions wie in „Meine Tasks"

„Alle Tasks im Tenant"

- Erfordert `tasks.read_all`
- Zusätzliche Spalte: Assignee
- Zusätzlicher Filter: Assignee (Multi-Select Person/Group)

Block D — Detail-View

`/{tenant}/tasks/:id`

Zu zeigende Daten:
- Titel
- Beschreibung (Markdown gerendert)
- Status, Priorität, Deadline
- Assignee (Person oder Group)
- Verknüpfte Entität (mit Klick-Link auf deren Detail-View)
- Ersteller + Erstellt-Datum
- Quelle (manuell / Chat-Nachricht mit Link / Form-Submission mit Link / Automation mit Link auf Automation-Rule)
- Bei `done`: Erledigt-Datum + Erledigt-von
- Watcher-Liste (Phase 1 minimal)
- Activity-Stream der Task (Status-Wechsel, Assignee-Wechsel, Deadline-Wechsel, Kommentare)

Benötigte UI-Elemente:
- Anzeige aller Stammdatenfelder mit Inline-Edit-Funktion
- Markdown-Editor für Beschreibung
- Date-Picker für Deadline
- Picker für Assignee (Person- oder Group-Suche, wechselbar)
- Picker für verknüpfte Entität (Auto-Complete über Contact/Company/Lead/Deal/Project)
- Status-Wechsler (`open → in_progress → done` , zurück möglich)
- Action-Funktionen: Erledigen, Wieder öffnen, Duplizieren, Löschen, Watcher hinzufügen/entfernen
- Activity-Stream-Komponente (Spec siehe Modul 1, Block E)
- Kommentar-Funktion (optional in Phase 1 — als reiner Notiz-Activity-Eintrag realisierbar)

Operationen:
- Stammdaten editieren (Titel, Beschreibung, Priorität, Deadline)
- Status ändern
- Assignee wechseln (Person ↔ Group)
- Verknüpfung anlegen / entfernen / ändern
- Watcher hinzufügen / entfernen
- Duplizieren (kopiert Stammdaten, neue Task im Status `open`)
- Löschen (Hard-Delete, mit Klartext-Warnung)

Klartext-Hinweise:
- Deadline überfällig: Banner *„Diese Aufgabe ist seit X Tagen überfällig"*
- Assignee deaktiviert: *„Zugewiesener Mitarbeiter ist nicht mehr aktiv. Bitte neu zuweisen."*
- Bei `done` -Wechsel ohne Auto-Archivierung-Hinweis: *„Erledigt — wird in 30 Tagen automatisch archiviert"*

Block E — Erstellungs-Quellen

E.1 — Manuelle Anlage

Eintritts-Punkte:
- Sidebar → *Aufgaben* → „Neue Aufgabe"
- In Entity-Detail-View (Contact/Company/Lead/Deal/Project) → Action „Aufgabe anlegen" (Pre-Fill `linked_entity_*`)
- In Listen-View → Quick-Add-Funktion (nur Titel, Rest später ergänzbar)

Anlage-Modal/-Page:
- Pflichtfeld: Titel
- Optionale Felder: Beschreibung, Assignee (Default = aktueller User), Deadline, Priorität (Default `normal`), Verknüpfung
- Klartext bei fehlendem Assignee: *„Wenn niemand zugewiesen ist, wirst automatisch du als Verantwortlicher eingetragen."*

E.2 — Aus Chat-Nachricht („Mark as Task")

- In Modul 13 (Team-Chat): Kontext-Menü pro Nachricht → „Als Aufgabe markieren"
- Modal öffnet sich mit Pre-Fill:
- Titel = erste 80 Zeichen der Chat-Nachricht (editierbar)
- Beschreibung = volle Chat-Nachricht (editierbar)
- Verknüpfung = Projekt/Deal/Contact, dem der Chat-Channel zugeordnet ist (falls vorhanden)
- Assignee = aktueller User (editierbar)
- `source = chat_message` , `source_ref = { chat_message_id }`
- Activity-Eintrag im Chat: *„[User] hat eine Aufgabe aus dieser Nachricht erstellt: [Titel]"* mit Link auf Task

E.3 — Aus Form-Submission

- Form-Action (Modul 5): „Aufgabe erstellen" als konfigurierbare Form-Action

- Konfiguration im Form-Builder: Assignee (User oder Group), Priorität, Deadline-Berechnung (z.B. „heute + 3 Tage"), Titel-Template (Platzhalter aus Form-Feldern), Beschreibungs-Template, Verknüpfung
- Bei Submission: Task wird angelegt mit `source = form_submission`, `source_ref = { form_submission_id }`
- Activity-Eintrag bei der Form-Submission

E.4 — Aus Automation-Action

- Automation-Engine (Modul 17) bietet Action-Typ: „Aufgabe erstellen"
- Konfigurierbare Parameter analog Form-Action
- `source = automation`, `source_ref = { automation_rule_id, automation_execution_id }`
- Klartext-Trail in Automation-Logs nach LP-10

Block F — Reminder & Auto-Archivierung

F.1 — Reminder-Notifications (an Modul 22)

Trigger-Zeitpunkte (pro Task mit `deadline_at`):
- 24 Stunden vor Deadline (Push + In-App)
- Bei Deadline (Push + In-App)
- 24 Stunden nach Deadline, falls noch `open` / `in_progress` (Push + In-App + Email als Eskalation)

Empfänger:
- Bei Person-Zuweisung: Assignee
- Bei Group-Zuweisung: alle Group-Mitglieder
- Plus Watcher

Konfigurierbarkeit pro User (Modul 22):
- Toggle „Task-Erinnerungen erhalten" (Default an)
- Ruhezeiten respektieren (Standard-Spec aus Modul 22)
- Anti-Spam-Bündelung: max. 1 Reminder-Notification pro 4h pro User, mehrere Tasks gebündelt

Klartext-Beispiele:
- Pre-Reminder: *„Die Aufgabe ‚Angebot an Schmidt schicken' ist morgen fällig"*
- At-Reminder: *„Die Aufgabe ‚Angebot an Schmidt schicken' ist heute fällig"*
- Post-Reminder: *„Die Aufgabe ‚Angebot an Schmidt schicken' ist seit 1 Tag überfällig"*

F.2 — Auto-Archivierung

- Cron-Job täglich (z.B. 03:00 Tenant-Zeitzone)
- Bedingung: `status = done` AND `completed_at <= now() - 30 Tage` AND `archived_at IS NULL`
- Setzt `archived_at = now()`
- Tasks bleiben in DB, sind nur per Filter „Archivierte anzeigen" sichtbar
- Hard-Delete erst manuell oder nach DSGVO-Workflows (Modul 35)

Klartext bei Übergang zu `done`:
- *„Aufgabe erledigt — wird in 30 Tagen automatisch archiviert"*

Block G — Activity-Stream-Integration

Erzeugte Activity-Typen (Spec aus Modul 1, Block E)

- `task_created` — bei Anlage (mit Hinweis auf Quelle)

- `task_assigned` — bei Wechsel des Assignee (mit von/zu)
- `task_completed` — bei Übergang zu `done`
- `task_reopened` — bei Übergang `done → open/in_progress`
- `task_deadline_changed` — bei Deadline-Update
- `task_priority_changed` — bei Priorität-Update
- `task_archived` — bei Auto-Archivierung

Verknüpfungen:
- Activity-`subject` ist die Task selbst
- `related_entity_refs` enthält die verknüpfte Entität (Contact/Deal/Projekt/...)
- Damit erscheint die Task-Activity automatisch im Stream der verknüpften Entität

Klartext-Summary-Beispiele (DE):
- *„Max Mustermann hat die Aufgabe ‚Angebot an Schmidt schicken' erstellt"*
- *„Lisa Müller hat die Aufgabe von Max Mustermann zu sich genommen"*
- *„Max Mustermann hat die Aufgabe ‚Angebot an Schmidt schicken' erledigt"*
- *„Automation ‚Lead-Eingang' hat eine Aufgabe für Vertrieb-Team erstellt"*

Block H — Permissions-Anwendung

Permission-IDs

Permission	Erlaubt
`tasks.read_own`	Eigene Tasks sehen (Assignee = self ODER Ersteller = self ODER Watcher = self ODER Group-Mitglied bei Group-Zuweisung)
`tasks.read_all`	Alle Tasks im Tenant sehen
`tasks.write`	Tasks anlegen + bearbeiten (Permissions-aware: nur eigene wenn nur `read_own`)
`tasks.admin`	Alle Tasks bearbeiten/löschen, Bulk-Operations, Auto-Archivierungs-Settings

Bundle-Mapping (Variante 2)

Bundle	Atomare Permissions
`tasks.none`	(keine)
`tasks.read_own`	`tasks.read_own`
`tasks.read_all`	`tasks.read_own` + `tasks.read_all`
`tasks.write`	`tasks.read_all` + `tasks.write`
`tasks.admin`	`tasks.read_all` + `tasks.write` + `tasks.admin`

Anwendung in Routes

- `/{tenant}/tasks` → `tasks.read_own` oder `tasks.read_all`
- `/{tenant}/tasks/new` → `tasks.write`
- `/{tenant}/tasks/:id` → `tasks.read_own` (mit Owner-Check) oder `tasks.read_all`
- Embed in Entity-Detail-View → entspricht den Tasks-Permissions zusätzlich zu den Entity-Permissions

Anwendung in Action-Buttons

- Anlegen: `tasks.write`
- Eigene Task bearbeiten: `tasks.write`
- Fremde Task bearbeiten (Assignee ≠ self UND Ersteller ≠ self): `tasks.write` + `tasks.read_all` ODER `tasks.admin`
- Löschen: `tasks.admin` ODER (`tasks.write` UND Ersteller = self)
- Bulk-Operations: `tasks.admin`

Owner-Check-Logik (für `read_own`)

User „besitzt" eine Task wenn:
- `assigned_to_user_id = self`, ODER
- User ist Mitglied der zugewiesenen Group (`assigned_to_group_id`), ODER
- `created_by_user_id = self`, ODER
- User ist Watcher (`TaskWatcher`)

Klartext-Fehler

- Zugriff auf nicht-eigene Task mit `tasks.read_own`: 404 (nicht 403)
- Edit-Versuch ohne Permission: Action-Buttons werden gar nicht gerendert
- Bulk-Action ohne `tasks.admin`: ausgegraut mit Tooltip *„Hierfür benötigst du erweiterte Berechtigungen"*

Block I — Klartext-Hinweise (zentral)

- Leere Task-Liste (Default „Meine Tasks"): *„Keine offenen Aufgaben — schöne Sache."*
- Leere gefilterte Liste: *„Mit den aktuellen Filtern keine Aufgaben gefunden. [Filter zurücksetzen]"*
- Task ohne Verknüpfung: kein Hinweis (legitim)
- Task mit gelöschter Verknüpfung: *„Die verknüpfte [Entität] wurde gelöscht. [Verknüpfung entfernen]"*
- Auto-Archivierungs-Hinweis siehe Block F
- Reminder-Texte siehe Block F

Phase 2 / spätere Features

- **Task-Templates:** wiederverwendbare Task-Definitionen (Titel, Beschreibung, Priorität, Deadline-Offset, Sub-Tasks) — pro Tenant verwaltet, in Anlage-Modal auswählbar
- **Wiederkehrende Tasks (cron):** Recurrence-Pattern (täglich/wöchentlich/monatlich/cron-Expression), erzeugt automatisch neue Task-Instanzen
- **Kanban-View:** zusätzliche Listen-Ansicht mit Spalten pro Status, Drag-&-Drop für Status-Wechsel
- **Dependencies:** Task X blockiert Task Y; Y wird erst sichtbar/aktiv wenn X erledigt
- **Sub-Tasks:** Eltern-Kind-Hierarchie mit Auto-Aggregation (Eltern erledigt wenn alle Kinder erledigt)
- **Time-Tracking pro Task:** Start/Stop, Summe in Stunden
- **Erweiterte Watcher-Logik:** automatische Watcher (z.B. Projektleiter bei Tasks im Projekt)
- **Task-Kommentare als eigene Sub-Entität** (statt nur Activity-Notizen)

Architektur-Vorschriften (Querschnitt)

- **AV #5 — Permissions-aware-Default:** Listen filtern automatisch nach User-Permissions; bei Realtime-Wechsel des Assignee verschwindet die Task aus der Liste fremder User
- **AV #6 — Audit-Trail-Default:** alle Statuswechsel, Assignee-Wechsel, Deadline-Wechsel als Activity geloggt
- **LP-2 — Büro vs. Baustelle:** Task-Liste muss in beiden UI-Modi funktionieren; Monteur sieht „Meine Tasks" prominent, Quick-Erledigen-Funktion mobil-optimiert
- **LP-9 — Spec-Tiefe:** UI-Anforderungen als Liste, kein Layout
- **LP-10 — Klartext-Diagnostik:** alle User-facing Texte fest formuliert (siehe Block I + F)
- **AV #20 — Action-Boundary für KI:** KI darf Tasks vorschlagen, nicht direkt anlegen ohne Bestätigung (relevant für Phase-2-KI-Vorschläge)

Detail-Spec-Status: DRAFT

Erste vollständige Detail-Spec für Modul 26. **Vor Bau:** Verifizierung mit Noah. Bei Implementierung dann konkrete Index-Definitionen, Migrations-SQL, Reminder-Cron-Konfiguration ergänzen.

Offene Detail-Punkte (nicht-blockierend):
- Konkrete Reminder-Zeitpunkte (24h/0h/+24h) ggf. tenant-konfigurierbar?
- Auto-Archivierungs-Frist (30 Tage) ggf. tenant-konfigurierbar?
- Watcher-Logik-Tiefe in Phase 1 (minimal vs. ausgebaut)
- Subframe-UI-Designs (separat)

Abhängige Module:
- Modul 13 (Team-Chat) — „Mark as Task"-Funktion
- Modul 5 (Form-Builder) — Form-Action „Aufgabe erstellen"
- Modul 17 (Automation-Engine) — Action „Aufgabe erstellen"
- Modul 22 (Notifications) — Reminder-Versand
- Modul 1 (CRM) / Modul 3 (Sales) / Modul 4 (Projekte) — Verknüpfungen + Embed-Anzeige