```
module: 20
title: Dashboard / Reporting
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-3, LP-9, LP-10]
spec_style: requirements-pur
```

Modul 20 — Dashboard / Reporting

> ***Detail-Spec gemäß LP-9.*** *Aggregierte KPIs aus allen anderen Modulen. Pro Rolle konfigurierbar.*

Themenblöcke

```
A. Datenmodell (Dashboard, DashboardLayout, Widget, WidgetInstance)
B. Menü-Struktur und Routes
C. Widget-Katalog (verfügbare Widget-Typen + Datenquellen)
D. Dashboard-Layout (User-Customizing)
E. Default-Layouts pro Rolle
F. Permissions-Anwendung
G. Performance & Realtime-Updates
```

Block A — Datenmodell

```
Dashboard           – Dashboard-Definition pro Tenant (mehrere möglich)
DashboardLayout     – User-spezifisches Layout pro Dashboard
WidgetCatalog       – Plattform-weite Widget-Definitionen (Code-Registry, nicht DB)
WidgetInstance      – konkrete Widget-Anwendung in einem Layout
```

Dashboard

- `id`, `tenant_id`, `name`, `description`
- `is_default_for_role` (optional Enum: `gf | sales_lead | sales_rep | pl | office | marketing | monteur`)
- `is_shared` (Boolean — gemeinsames Tenant-Dashboard vs. nur User-spezifisch)
- `created_by_user_id`, `archived_at`

DashboardLayout

- `id`, `tenant_id`, `dashboard_id`, `user_id` (NULL bei `is_shared=true`)
- `widgets[]` (JSON-Array von WidgetInstance-IDs mit Position + Größe)
- `layout_data` (JSON — Grid-Position pro Widget)
- `updated_at`

Logik:
- Pro User pro Dashboard ein Layout (User passt individuell an)
- Falls Dashboard `is_shared=true`: ein gemeinsames Layout, von Tenant-Admin verwaltet

WidgetInstance

- `id`, `tenant_id`, `widget_catalog_key`
- `config` (JSON — Widget-spezifische Konfiguration: Filter, Zeitraum, Anzeige-Optionen)
- `position`, `size`

WidgetCatalog (Code-Registry)

Plattform-weit definierte Widget-Typen:
- Schlüssel + Default-Config-Schema
- Render-Komponente
- Datenquelle (welche Module / welche Permissions)
- Erweiterbar pro Modul

Block B — Menü-Struktur und Routes

```
/{tenant}/                              → Default-Dashboard (nach Login)
/{tenant}/dashboard                     → expliziter Dashboard-Pfad
/{tenant}/dashboard/:id                 → Spezifisches Dashboard (bei Multi-Dashboard-Tenants)
/{tenant}/dashboard/:id/edit            → Layout-Edit-Modus

/{tenant}/admin/dashboards              → Tenant-Dashboard-Verwaltung
/{tenant}/admin/dashboards/:id          → Editor für shared Dashboard
```

Default-Routing-Logik:
1. User hat persönliches Layout für ein Dashboard → das wird gezeigt
2. Sonst: Dashboard mit `is_default_for_role` matchend zur User-Rolle
3. Sonst: Tenant-Default-Dashboard (shared)
4. User-Switcher (oben in Dashboard-View): zeigt verfügbare Dashboards

---

Block C — Widget-Katalog

Sales (M3)

Widget-Key	Daten	Permission
`sales.pipeline_kpis`	Anzahl Deals × Stage, Aggregat-Volumen	`deals.read_aggregates`
`sales.deal_trend`	Anzahl/Volumen über Zeitraum	`deals.read_aggregates`
`sales.revenue_forecast`	Forecast (Volumen × Wahrscheinlichkeit)	`deals.read_aggregates`
`sales.salesperson_performance`	Pro Vertriebler: Deals, Volumen, Won-Rate, Aktivitäts-KPIs	`deals.read_aggregates`
`sales.lost_reasons`	Lost-Reason-Verteilung	`deals.read_aggregates`
`sales.stage_conversion`	Conversion-Rate pro Stage-Übergang	`deals.read_aggregates`
`sales.my_open_deals`	Eigene aktive Deals als Liste	`deals.read_own`

Projects (M4)

Widget-Key	Daten	Permission
`projects.pipeline_kpis`	Anzahl Projects × Stage, Volumen	`projects.read_aggregates`
`projects.project_overview`	Status-Verteilung	`projects.read_aggregates`
`projects.utilization`	MA-Auslastung	`plantafel.read_all` + `projects.read_aggregates`
`projects.open_defects`	Offene Mängel aus Acceptance	`projects.read_all`
`projects.my_active_projects`	Eigene aktive Projekte	`projects.read_own_member`

Tasks (M26)

Widget-Key	Daten	Permission
`tasks.my_tasks`	Eigene offene Tasks (heute/Woche/überfällig)	`tasks.read_own`
`tasks.team_completion`	Team-Task-Completion-Rate	`tasks.read_all`

Activity (M1)

Widget-Key	Daten	Permission
`activity.my_recent`	Eigene letzte Activities	`crm.read_*`
`activity.team_recent`	Team-Activities chronologisch	`crm.read_all`

Communication (M9, M12)

Widget-Key	Daten	Permission
`comms.unread_emails`	Anzahl ungelesene Mails	`email.read_*`
`comms.calls_today`	Anzahl Anrufe heute	`telephony.read_*`

Greetings & Status

Widget-Key	Daten	Permission
`general.greeting`	„Guten Morgen, Noah!" + Wetter optional	(alle)
`general.kwp_status`	KWP-Sync-Status, letzte Sync	`tenant.admin`
`general.tenant_health`	Subscription-Status, Usage-Anzeige	`tenant.admin`

HR (M14)

Widget-Key	Daten	Permission
`hr.team_today`	Wer ist heute da / Urlaub / krank	`hr.read_*`
`hr.pending_requests`	Offene Genehmigungs-Anträge	`hr.approve_*`

AI (M18)

Widget-Key	Daten	Permission
`ai.suggestions`	KI-Vorschläge basierend auf Tenant-Kontext	`ai.read`

Block D — Dashboard-Layout (User-Customizing)

Edit-Modus (`/{tenant}/dashboard/:id/edit`)

Benötigte UI-Elemente:
- Grid-Layout-Komponente (Drag-&-Drop + Resize)
- Widget-Catalog-Picker (gegated nach User-Permissions)
- Pro Widget-Instance: Konfigurations-Funktion (Widget-spezifisch)
- Funktionen: Widget hinzufügen / verschieben / resizen / löschen
- Save-Funktion
- Reset-Funktion (zurück zum Role-Default)

Operationen:
- Widget hinzufügen aus Catalog
- Position + Größe ändern
- Widget-Config editieren
- Widget entfernen
- Dashboard duplizieren (als Vorlage für anderen Dashboard)
- Dashboard löschen

Klartext-Hinweise:
- Widget ohne Permission im Catalog ausgegraut: *„Hierfür benötigst du erweiterte Berechtigungen"*
- Layout-Reset: Bestätigungs-Dialog *„Dein angepasstes Layout wird durch das Standard-Layout deiner Rolle ersetzt."*

Block E — Default-Layouts pro Rolle (SHK-Template)

Beim Tenant-Anlegen werden 7 Role-Default-Dashboards geseedet (`is_default_for_role` gesetzt, `is_shared=true`). Alle Default-Dashboards sind **deklarativ als Branchen-Template** gespeichert (LP-3 + AV #3).

GF-Dashboard

`general.greeting` , `sales.pipeline_kpis` , `sales.revenue_forecast` (3-Monats), `sales.salesperson_performance` , `projects.pipeline_kpis` , `projects.utilization` , `projects.open_defects` , `general.kwp_status` , `general.tenant_health` , `hr.team_today`

Vertriebsleiter-Dashboard

`general.greeting` , `sales.pipeline_kpis` , `sales.deal_trend` (Monats-Trend), `sales.salesperson_performance` , `sales.lost_reasons` , `sales.stage_conversion` , `tasks.team_completion`

Vertriebler-Dashboard

`general.greeting` , `sales.my_open_deals` , `tasks.my_tasks` , `comms.unread_emails` , `comms.calls_today` , `activity.my_recent`

Projektleiter / Bauleiter-Dashboard

`general.greeting`, `projects.my_active_projects`, `projects.open_defects` (eigene), `projects.utilization` (eigenes Team), `tasks.my_tasks`, `hr.team_today`

Büro / Backoffice-Dashboard

`general.greeting`, `projects.pipeline_kpis`, `tasks.my_tasks`, `comms.unread_emails`, `hr.pending_requests`, `general.kwp_status`

Marketing-Dashboard

`general.greeting`, `sales.deal_trend`, `sales.lost_reasons`, `sales.stage_conversion`

Monteur-Dashboard (Mobile-First, LP-2)

„Meine Termine heute" (eigene ScheduleEntries), *„Meine aktiven Projekte"*, *„Meine Tasks"*, optional KI-Walkthrough-Hinweise

---

Block F — Permissions

Permission	Erlaubt
`dashboard.read`	Default-Dashboard sehen
`dashboard.customize`	Eigenes Layout anpassen
`dashboard.create`	Neues persönliches Dashboard anlegen
`dashboard.share`	Dashboard als shared markieren
`dashboard.admin`	Tenant-Admin-Verwaltung (Default-Layouts pro Rolle, shared Dashboards)

Bundle-Mapping

Bundle	Atomare Permissions
`dashboard.basic`	`dashboard.read` + `dashboard.customize`
`dashboard.power`	obigen + `dashboard.create`
`dashboard.admin`	alle

Default-Rollen-Zuweisung

Rolle	Permission
GF	`dashboard.admin`
Vertriebsleiter / PL / Büro	`dashboard.power`
Vertriebler / Marketing / Monteur	`dashboard.basic`

Widget-Sichtbarkeit

- Pro Widget gilt die Permission des Quellen-Moduls (z.B. `sales.pipeline_kpis` braucht `deals.read_aggregates`)
- User ohne Permission sieht das Widget nicht im Catalog (Permissions-aware-Default)

- Bei Rollen-Wechsel mit verlorenen Widget-Permissions: Widgets werden ausgeblendet mit Klartext im Edit-Modus

---

Block G — Performance & Realtime-Updates

Caching-Strategie

- Pro Widget-Instance Cache mit TTL (Default 60 Sek, Widget-spezifisch konfigurierbar)
- Cache-Schlüssel: `tenant_id + widget_key + config_hash + user_id`
- Cache-Invalidierung bei relevanten Änderungen

Realtime-Updates

- WebSocket-Push an aktive Dashboard-Sessions bei wichtigen Änderungen (Deal-Won, neuer Lead, Mängel-Acceptance)
- Subtle-Animation bei Update (Designer-Detail)
- Opt-Out-Option pro User: *„Live-Updates pausieren"*

Lazy Loading

- Widgets werden lazily geladen (nur sichtbare im Viewport)
- Skeleton-State first-render

Aggregat-Berechnung

- Server-side, nicht client-side
- Bei großen Datasets: Pre-Calculated-Aggregat-Tables, asynchron via Cron erneuert
- Klartext bei veralteten Aggregaten: *„Daten zuletzt aktualisiert vor X Minuten"*

Architektur-Vorschriften (Querschnitt)

- **LP-2:** Monteur-Dashboard ist mobile-first
- **LP-3, AV #3:** Default-Dashboards sind deklaratives Branchen-Template
- **AV #5:** Permissions-aware-Default — Widgets gegated nach Quellen-Modul-Permissions
- **AV #6:** Audit-Trail für Layout-Änderungen (wer hat shared Dashboard geändert)

Detail-Spec-Status: DRAFT

Modul 20 komplett spec'd.

Offene Detail-Punkte:
- Konkrete Widget-Konfigurations-Schemas pro Widget-Typ (Implementierungs-Detail)
- Grid-Layout-Library-Wahl (Boilerplate-Entscheidung, z.B. react-grid-layout)
- WebSocket-Strategie (gemeinsam mit M13 Chat)
- KI-Widget-Verhalten konkretisieren (M18-Spec-Kontext)
- Konkrete Aggregat-Cache-Invalidierungs-Strategie