Roadmap: HiveGrid 2.0

Overview

HiveGrid 2.0 entsteht in einem Big-Bang-Launch (AD-19): erst live, wenn das vollständige Phase-1-Produkt steht. Die interne Bau-Reihenfolge folgt der Hybrid-Strategie AD-11 — zuerst ein Walking-Skeleton-Fundament (Boilerplate einziehen, EU-Sterilisation, Integration-Contracts **plus die Service-Kerne Audit, Notifications, i18n und Feature-Flags** — sie sind Vorbedingung des gesamten Auth-Blocks), dann Identität/Multi-Tenancy, die Querschnitts-Oberflächen (Audit-Viewer, Notification-Center, Sprach-UX) und die nutzerseitigen Primitive (Files, Tasks), darauf die Apps-Engine, das CRM-Datenmodell und die durchgängige SHK-Wertschöpfungskette Lead → Sales → Auftrag (KWP-Sync = erstes v2-Modul — API-Zugang erst ~19.07., nach v1-Launch; Scope-Schnitt #12). Erst danach kommt die Breite: globale Suche/Dashboard, die USP-Säulen Automation-Engine und KI-Layer (AD-16), die Kommunikations-Bausteine (Email/Marketing, Telefonie/Chat, Kalender/Booking/Forms/Signatur), HR-Hub, Customer-Portal und schließlich die Plattform- und Launch-Schicht (Admin, Billing/Onboarding/Branding, Public-API/DSGVO/Mobile/Help). Jede Phase liefert eine eigenständig verifizierbare Fähigkeit; Coverage über alle 90 v1-Requirements ist lückenlos (Scope-Schnitte 11./12.06.2026: 19 Reqs → v2). Die Kommunikations-Phasen hängen primär an der Apps-Engine (Connections) und am CRM (Contact-Matching) — sie brauchen NICHT die Automation/KI-Säulen als Vorbedingung. **Dependency-Inversion-Prinzip:** Es existieren keine unaufgelösten Vorwärts-Abhängigkeiten — spätere Module integrieren sich rückwirkend in frühere Oberflächen (Ownership-Register + Extension-Point-Pflichten in `PHASE-SOURCES.md`); jeder plan-phase-Lauf MUSS die Retro-Integrations-Aufgaben seiner Phase aus dem Register mit einplanen.

> 🔴 ***MODELL-PIVOT 2026-06-13 (Projekt-zentrisch, AD-20):*** *Das CRM/Pipeline-Modell wird umgebaut — ein* `Projekt` *-Kern, Pipelines = konfigurierbare Views; „Deal"/„Auftrag" = Pipeline-Typen, keine eigenen Entitäten;* ***Lead → Projekt****; Abnahme bleibt v1 als Checklisten-Sub-Area; Rechnung = Sub-Entity v1. Betrifft* ***P6–P9*** *(siehe Phasen-Notizen unten). Die Success-Criteria/Requirements von P6–P9 werden gemeinsam mit dem neuen ER-Modell neu gefasst. Single Source:* `.planning/ENTITY-MODEL.md` .

Phases

Phase Numbering:

- Integer phases (1, 2, 3): Planned milestone work
- Decimal phases (2.1, 2.2): Urgent insertions (marked with INSERTED)

Decimal phases appear between their surrounding integers in numeric order.

- [ ] **Phase 1: Fundament & Walking-Skeleton** - Boilerplate einziehen, EU-Sterilisation, Querschnitts-Neubauten + Service-Kerne (Audit, Notifications, i18n, Feature-Flags), Integration-Contracts als Bau-Vertrag
- [ ] **Phase 2: Identität & Mandantentrennung** - Auth, Multi-Tenancy, RLS-Baseline, RBAC, Sessions (API-Keys → v2 mit Public-API)
- [ ] **Phase 3: Querschnitts-Oberflächen** - Audit-Viewer/Export, Notification-Center/Settings, Sprach-Settings DE-only (Service-Kerne liegen in Phase 1)
- [ ] **Phase 4: Files & Tasks** - Nutzerseitige Plattform-Primitive: Datei-Verwaltung (chunked Upload, Preview inkl. HEIC→JPG) und Tasks
- [ ] **Phase 5: Apps-Engine & Integrations-Framework** - DataConnection, Adapter-Registry, OAuth/API-Key-Connect-Lifecycle
- [ ] **Phase 6: CRM-Kern** - Typisiertes CRM-Datenmodell, Listen/Detail-Views, Custom-Felder, Tags, Activity-Stream, Duplikate
- [ ] **Phase 7: Lead-Inbox** - Lead-Triage-Arbeitsplatz, Multi-Quellen-Auto-Anlage, Anti-Spam, Statistiken
- [ ] **Phase 8: Sales-Pipeline** - Multi-Pipeline-Deals, Kanban, Forecast/KPIs, Field-Level-Permissions
- [ ] **Phase 9: Auftrags-/Projektpipeline** - Projekt-Pipelines, Plantafel, digitale Abnahme (offline-fähig)
- [ ] ~~**Phase 10: KWP-Integration**~~ — **→ v2, ERSTES v2-Modul** (deferred 2026-06-12; API-Zugang ~19.07. > Launch; Nummer reserviert, Ausführung überspringt P10)
- [ ] **Phase 11: Globale Suche & Dashboard** - Keyword-Suche (Meilisearch, AD-5 sequenziert), Cmd+K, Widget-Dashboard mit festen Rollen-Layouts

- [ ] **Phase 12: Automation-Engine** - Visueller Flow-Editor, Trigger/Action-Library, Guards, Sprach-zu-Automation
- [ ] **Phase 13: KI-Layer** - EU-LLM-Gateway, Multi-Provider-Fallback, Promptfoo, Sprach-zu-Automation-Unterbau (HiveMind/RAG/Whisper → v2, Andock-Adapter in v1)
- [ ] **Phase 14: Email & Marketing** - Sende-Domain-Verifikation, BCC-Logging-Brücke, DSGVO-konforme Marketing-Kampagnen (Postfach-Sync → v2)
- [ ] **Phase 15: Telefonie & Team-Chat** - AirCall-Click-to-Call/Inbound, Real-Time-Team-Chat mit entity-linked Channels
- [ ] **Phase 16: Kalender, Booking, Forms & Signatur** - Kalender-Sync, öffentliche Buchungsseiten, Form-Builder, Angebots-Signatur
- [ ] **Phase 17: HR-Hub** - Verschlüsselter HR-Hub, Passwort-Safe, Urlaubs-/Krank-Workflow
- [ ] ~~**Phase 18: Customer-Portal**~~ — **→ v2** (deferred 2026-06-11; Nummer reserviert, Ausführung überspringt P18)
- [ ] **Phase 19: Plattform-Admin** - Getrennte Admin-Welt (admin.hivegrid.de), Tenant-Management + Abo-Zusammenstellung, Queue-Monitoring, Support-Inbox
- [ ] **Phase 20: Billing, Onboarding & Branding** - Stripe-Billing/Trial/Dunning (Abo-Baukasten-Recut), Auto-SHK-Provisionierung, Theme-Preference
- [ ] **Phase 21: DSGVO & Launch-Reife** - DSGVO-Export + Lösch-Workflow (Crypto-Shredding), Feedback-Widget, Launch-Checkliste

v2-Phasen (Sprint Mo 13.07.–Fr 31.07., nach Launch — 43 Milestones, 2 Tracks dependency-frei parallel)

> 🟦 ***v2 = committed-deferred (kein Vielleicht-Backlog).*** *Voll ausgearbeitete Detail-Fassung (Success-Criteria · Screens · v1-Andockpunkte · Recut/Risiko) je Phase:* `.planning/ROADMAP-V2.md` *. Bei den „phase:1"-Modulen (P26/P28/P29/P30) ist nur das* ***v2-Delta*** *gemeint, nicht das v1-Fundament.*

Block 1 — sofort baubar (alle Infos vorhanden):

- [ ] **Phase 22: Public-API & Webhooks** - Versionierte Read-REST-API + signierte Webhooks + Tenant-API-Keys mit Scopes/Rate-Limit
- [ ] **Phase 23: Postfach-Sync (Email-Inbox)** - Bidirektionale Gmail/Outlook-Inbox mit CRM-Matching, Threading, Versand über User-Mailbox
- [ ] **Phase 24: Self-Service-Billing & Onboarding-Wizard** - Self-Service-Abo-Zusammenstellung (Web-Checkout) + geführter branchen-getriebener Wizard
- [ ] **Phase 25: KI-Tiefe — HiveMind/RAG** - RAG über Tenant-Daten, EU-Whisper, pgvector-Hybrid-Suche, autonomer HiveMind-Agent (Action-Boundary)
- [ ] **Phase 26: Customer-Portal** ⭐ - Endkunden-Portal (eigene Auth/Magic-Link, Projekt-Sicht, Upload, Reklamation) — **kritisch** (ex-P18)
- [ ] **Phase 27: Mobile-PWA-Ausbau** - Installierbare PWA (Add-to-Homescreen, Service-Worker-Offline, Web-Push) für den Baustellen-Einsatz — kein nativer Client/App-Store (AD-17, PWA-only)
- [ ] **Phase 28: Plattform-Admin-Tiefe** - Getrennte AdminUser-Identität/AdminRBAC + Feature-Flag-Massen-Statistik + interne Lead-Pipeline
- [ ] **Phase 29: Help-Center & Help-Desk** - Öffentliches Help-Center (SEO, kontextuell) + Email-Ticket-Desk + Feedback-Ausbau
- [ ] **Phase 30: Dashboard-Deluxe & Querschnitt** - Custom-Widgets/Export + Notif-Bündelung/Quiet-Hours + EN-Locale (DeepL)
- [ ] **Phase 31: DSGVO-Vollausbau** - AVV-Versionierung + Subprozessoren-Workflow + Auskunfts-/Voll-Export-Workflow (ersetzt manuelles Playbook)

Block 2 — Partner-API-Vertrag offen → ans Ende, agil vorziehbar, sobald Zugang/Spec da:

- [ ] **Phase 32: KWP-Integration** *(Recut)* - Bidirektionaler Sync HiveGrid↔KWP über die offizielle REST-API (ex-P10; Zugang ~20.07.; kwpsync.exe NICHT bauen)
- [ ] **Phase 33: Telefonie-Erweiterung (HalloPetra)** - HalloPetra als zweiter Provider hinterm TelephonyProviderInterface (⚠ Spec offen)

- [ ] **Phase 34: HR-Drittanbieter-Onboarding** - Memberspot/Boobora via Apps-Engine (⚠ Spec offen)

Phase Details

Phase 1: Fundament & Walking-Skeleton

Goal: Das Repo ist eine lauffähige Codebasis, EU-steril, mit den Querschnitts-Neubauten, den Service-Kernen (Audit, Notifications, i18n, Feature-Flags) und den verbindlichen Integration-Contracts als Bau-Vertrag für alle Folgephasen. **Depends on**: Nothing (first phase) **Requirements**: REQ-integration-sidebar, REQ-integration-routes, REQ-integration-activity-wiring, REQ-integration-audit-wiring, REQ-integration-notification-wiring, REQ-integration-view-model-pflicht, REQ-integration-smoke-tests, REQ-audit-datenmodell, REQ-notification-datenmodell, REQ-i18n-architektur, REQ-feature-flags-system **Source-Specs**: `docs/ARCHITECTURE_DECISIONS.md` , `docs/INTEGRATION_CONTRACTS.md` ; Service-Kern-Teile aus `_bmad-output/specs/M21-audit-log.md` , `M22-notifications.md` , `M30-i18n.md` , `M32-feature-flags.md` **Milestone-PLANs**: 15 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 1) **Success Criteria** (what must be TRUE):

1. Die ToolCTRL-Boilerplate ist als Git-Downstream eingezogen, baut lokal durch und der upstream-Remote bleibt mergebar erhalten (AD-10). *Planner-Hinweis: Ist-Stand prüfen — laut ADR-Text „Docs-only", real ist die Codebase seit 2026-06-04 eingezogen; bereits Erledigtes nicht neu planen.*
2. GitHub/Google-OAuth-Reste und `User.githubId/googleId` sind entfernt; keine US-Kern-Abhängigkeit mehr im Auth-Pfad (EU-Sterilisation).
3. `logActivity()` , Search-Registry, Subframe-Adapter und Mass-Action-Schutz existieren als aufrufbare Querschnittsdienste mit Smoke-Test.
4. Die Service-Kerne existieren als aufrufbare Dienste mit Smoke-Test: AuditEvent mit SHA-256-Hash-Kette via Prisma-Middleware (default-on), `notify()` mit Kanal-Abstraktion (In-App/Email; Push-Kanal als Stub bis v2), `packages/i18n` mit Build-Lint gegen Hard-Strings und Resolution User → Tenant → DE, Feature-Flag-Engine mit `tenantHasFlag()` + Redis-Cache (Admin-UI folgt in P19).
5. **Entitlement-Gate-Pflicht (Konnektor-Prinzip):** Das Gate-Muster `tenantHasModule('<modul>')` existiert mit Beispiel-Anwendung an der Walking-Skeleton-Route — jede Modul-Phase registriert damit ihren Feature-Konnektor (Sidebar/Routes/Actions plan-gebunden schaltbar). Vorlage: `archive/hivegrid-cloud/app/config/pricing-plans.ts` .
6. Eine Beispiel-Route gibt im Loader ein View-Model zurück, ist in der Sidebar registriert und hat einen grünen Smoke-Test gegen `packages/ui-mock` .
7. Die verbindlichen URL-Schemata (App, Admin, Portal, Forms, Booking, Help, API) sind als geprüfte Konvention dokumentiert. **Plans**: TBD **UI hint**: yes

Phase 2: Identität & Mandantentrennung

Goal: Ein Handwerksbetrieb kann sich selbst registrieren, sicher anmelden und in einem vollständig RLS-isolierten Mandanten mit rollenbasierten Rechten arbeiten. **Depends on**: Phase 1 **Requirements**: REQ-auth-multi-tenancy, REQ-auth-methoden, REQ-auth-rbac, REQ-auth-user-lifecycle **Source-Specs**: `_bmad-output/specs/M27-auth-multi-tenancy.md` **Milestone-PLANs**: 9 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 2); Scope-Schnitt 12.06.: M027 (API-Keys) → v2 mit Public-API — Boilerplate- `ApiKey` -Modell bleibt ungenutzt im Code **Success Criteria** (what must be TRUE):

1. Ein Nutzer kann sich per Self-Service registrieren, erhält einen auto-generierten Tenant-Slug (Reserved-Words geblockt) und landet unter app.hivegrid.de.
2. Login mit Email+Passwort (NIST/Breach-Check) plus TOTP-2FA funktioniert; Sessions sind in Postgres serverseitig (Geräte-Liste, Revoke, „von allen Geräten abmelden") mit Redis-Cache (AD-4).
3. RLS-Policies (USING/WITH CHECK, `SET LOCAL app.tenant_id` , App-Rolle ohne BYPASSRLS) verhindern nachweisbar mandantenübergreifenden Datenzugriff (AD-1).
4. 10 SHK-Standard-Rollen sind geseedet; ein Admin kann Nutzer einladen (48h-Token), deaktivieren und DSGVO-konform per PII-Anonymisierung löschen.
5. *(→ v2, Scope-Schnitt 12.06.)* ~~Ein Nutzer kann API-Keys mit Scopes und Rate-Limit erstellen~~ — wandert mit der Public-API (M161–163) nach v2. **Plans**: TBD **UI hint**: yes

Phase 3: Querschnitts-Oberflächen

Goal: Die Bedien-Oberflächen der in Phase 1 gebauten Plattformdienste stehen — Audit-Viewer mit Export, Notification-Center mit Nutzer-Einstellungen und die Sprach-Settings (v1 DE-only). **Depends on**: Phase 2 **Requirements**: REQ-audit-retention-export, REQ-audit-permissions, REQ-notification-versand-pipeline, REQ-i18n-outbound **Source-Specs**: `_bmad-output/specs/M21-audit-log.md`, `_bmad-output/specs/M22-notifications.md`, `_bmad-output/specs/M30-i18n.md` (Oberflächen-Teile; Service-Kerne → Phase 1) **Milestone-PLANs**: 5 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 3); Scope-Schnitt 12.06.: M029 (Sprach-Wechsler) + M117 (Bündelung/Quiet-Hours) → v2; M028/M152 gedünnt **Success Criteria** (what must be TRUE):

1. Ein Nutzer mit Audit-Permission sieht den Audit-Viewer (gefiltert, permission-gegated); Export als NDJSON/CSV ins Object Storage (Hetzner-S3) funktioniert mit fixer Default-Retention. *(Scope-Schnitt 12.06.: konfigurierbare Retention-Policy/Legal-Hold/Saved-Views → v2; Boilerplate liefert Retention-Cron + PDF-Export.)*
2. Die Versand-Pipeline stellt Notifications über BullMQ in alle Kanäle zu (In-App via SSE, Email; Push-Stub bis v2) — sofort, ohne Bündelung; ein Nutzer sieht das Notification-Center und schaltet Kategorien pro Kanal an/aus. *(Scope-Schnitt 12.06.: Quiet-Hours/Bündelung/Push-Limits = M117 → v2, bewusste Entscheidung.)*
3. Alle Oberflächen-Strings laufen durch `t()` mit vollständiger `de`-Locale (`verify:i18n`-Gate grün); Contact.language ist als Feld vorhanden und steuert die Outbound-Sprache ab v2. *(Scope-Schnitt 12.06.: v1 DE-only; Wechsler + Fallback-UX + EN-Templates → v2 — alles i18n-ready, Nachrüsten ohne Refactoring.)* **Plans**: TBD **UI hint**: yes

Phase 4: Files & Tasks

Goal: Die nutzerseitigen Plattform-Primitive stehen bereit — Dateien lassen sich an jede Entität hängen (chunked Upload, Preview inkl. HEIC→JPG) und Tasks mit Fälligkeit/Reminder an CRM-/Sales-/Projekt-Entitäten. **Depends on**: Phase 2 **Requirements**: REQ-files-datenmodell, REQ-files-upload-preview, REQ-tasks-datenmodell, REQ-tasks-reminder-archivierung **Source-Specs**: `_bmad-output/specs/M23-files.md`, `_bmad-output/specs/M26-tasks.md` **Milestone-PLANs**: 7 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 4); M074 (Task-ERP-Sync) → Phase 10; Scope-Schnitt 12.06.: M054/M055 gedünnt **Success Criteria** (what must be TRUE):

1. Ein Nutzer kann Dateien chunked hochladen (TUS) mit RLS auf Storage-Pfaden (S3-kompatibles Hetzner Object Storage, Infra-ADR-009); Previews (PDF→PNG, Bild→WebP, **HEIC→JPG — Pflicht für iPhone-Monteur-Fotos**) sind sichtbar. *(Scope-Schnitt 12.06.: Versionierung, Office-/Video-Previews, Resume-Upload, OCR → v2; Permissions/Sharing + Tags + Archiv-Flag bleiben.)*
2. Dateien lassen sich über FileEntityLink an Entitäten hängen; Customer-Portal-Upload ist vorbereitet.
3. Ein Nutzer kann Tasks mit Status, Priorität, Fälligkeit und Entitäts-Verknüpfung (Contact/Company/Lead/Deal/Project) anlegen, sich als Watcher eintragen und einen Reminder (via notify()) erhalten.
4. Abgeschlossene Tasks werden nach 30 Tagen per Cron auto-archiviert. **Plans**: TBD **UI hint**: yes

Phase 5: Apps-Engine & Integrations-Framework

Goal: Externe Systeme werden über ein einziges, typsicheres Integrations-Framework angebunden — der Vertrag, über den später Gmail, Outlook, AirCall, PandaDoc, Stripe und KWP laufen. **Depends on**: Phase 2 **Requirements**: REQ-apps-engine-datenmodell, REQ-apps-engine-lifecycle **Source-Specs**: `_bmad-output/specs/M33a-apps-engine.md` **Milestone-PLANs**: 3 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 5) **Success Criteria** (what must be TRUE):

1. Ein Admin kann eine DataConnection über einen OAuth2-Connect-Flow (Gmail, Outlook, AirCall, PandaDoc) oder API-Key-Connect (KWP, Stripe) in der Settings-UI herstellen.
2. Die type-safe Adapter-Registry erlaubt das Registrieren neuer Connection-Typen ohne Änderung am Kern; AppSyncRun/AppSyncItem protokollieren Sync-Läufe.
3. Connection-Health-Monitoring zeigt den Status jeder Verbindung; OAuth-Tokens sind per-Tenant-DEK-verschlüsselt gespeichert (AD-8). **Plans**: TBD **UI hint**: yes

Phase 6: CRM-Kern

> 🔴 ***Pivot-Note (AD-20, 13.06.):*** *Kontaktmodell wird umgebaut: `Kontakt` = Oberbegriff mit `Kontakttyp` (Privatperson/Organisation), `Kontaktstatus` (Kontakt/Kunde/Bestandskunde), `Kontaktart` (Haupt-/Sekundärkontakt als Rolle/Relation). Zwei Listen-Einstiege (Unternehmen/Organisationen + Kontakte). Custom-Fields + Pipeline-Feld-Config-Layer. SC/Reqs werden mit dem ER-Modell neu gefasst.*

Goal: Ein Vertriebler kann Kontakte, Firmen und Leads als typisierte First-Class-Entitäten verwalten — mit Custom-Feldern, Tags, Activity-Stream und Duplikat-Schutz. **Depends on**: Phase 5, Phase 4 **Requirements**: REQ-crm-datenmodell, REQ-crm-listen-views, REQ-crm-detail-views, REQ-crm-lead-promotion, REQ-crm-activity-stream, REQ-crm-custom-felder, REQ-crm-duplikat-erkennung, REQ-crm-tags, REQ-crm-permissions **Source-Specs**: `_bmad-output/specs/M01-crm-kern.md` **Milestone-PLANs**: 5 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 6) **Success Criteria** (what must be TRUE):

1. Alle CRM-Kernentitäten (Contact, Company, Lead, Relation, Activity, CustomField, Tag) sind typisierte Prisma-Tabellen mit RLS-Policies; der Boilerplate-Entity-Builder ist entfernt (AD-2).
2. Ein Nutzer kann Kontakt-/Firmen-Listen server-seitig filtern, sortieren, paginieren, per Bulk-Action bearbeiten und als CSV exportieren.
3. Eine Detailansicht zeigt Inline-Edit, Activity-Stream (append-only Event-Sourcing), verknüpfte Entitäten und Custom-Felder; SHK-Branchen-Custom-Felder sind als Template geseedet.
4. Beim Anlegen warnt ein Live-Match-Check vor Duplikaten; ein Lead kann per Promotions-Modal (Auto-Email-Match, 4 Optionen) transaktional zum Contact werden.
5. crm.*-Permission-Bundle (read_own/read_all/write/delete/admin) filtert serverseitig nach Owner; ein Nutzer ohne read_all sieht nur eigene Datensätze. **Plans**: TBD **UI hint**: yes

Phase 7: Lead-Inbox

> 🔴 ***Pivot-Note (AD-20, 13.06.):*** *Lead/LeadInput ist Vorstufe mit Datenpunkt* `Weiterverarbeitungsfreigabe` *(Ja/Nein/Unklar).* ***Lead → Projekt*** *statt Lead→Contact-Promotion (4-Optionen-Modal entfällt); Kontakt wird separat verknüpft. Vererbung relevanter Lead-Daten ins Projekt bei Freigabe. SC/Reqs werden mit dem ER-Modell neu gefasst.*

Goal: Eingehende Leads aus allen Quellen landen in einem Triage-Arbeitsplatz, werden automatisch angelegt, gegen Spam geschützt und sind auswertbar. **Depends on**: Phase 6 **Requirements**: REQ-lead-inbox-liste, REQ-lead-quellen-auto-anlage, REQ-lead-status-workflow, REQ-lead-anti-spam, REQ-lead-statistiken, REQ-lead-permissions **Source-Specs**: `_bmad-output/specs/M02-lead-inbox.md` **Milestone-PLANs**: 4 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 7); M049 (Auto-Verkaufsfall) → Phase 8 (braucht Deal-Entity); P7 liefert das `lead.qualified`-Event als Extension-Point **Success Criteria** (what must be TRUE):

1. Ein Bearbeiter sieht die Lead-Inbox mit Status-Filter, Quick-Actions, Bulk-Operationen und Real-time-Refresh.
2. Leads aus Form-Submission, eingehender Email und AirCall-Inbound werden automatisch angelegt und stoßen Automation-Trigger an (sofern Automation aktiv).
3. Der Status-Workflow (new|qualified|converted|lost|spam) erlaubt nur definierte Übergänge; ein DSGVO-Auto-Lösch-Cron räumt alte Leads ab.
4. Rate-Limiting/Anomalie-Detection schiebt verdächtige Leads in Quarantäne und benachrichtigt den Tenant-Admin.
5. Eine Statistiken-Page zeigt KPI-Kacheln, Quellen-Aufschlüsselung und Trends, permission-aware (leads.*-Bundle mit Owner-Check). **Plans**: TBD **UI hint**: yes

Phase 8: Sales-Pipeline → Projekt & Pipelines (Teil 1: Kern + Pipeline-Engine)

> 🔴 ***Pivot-Note (AD-20, 13.06.):*** *P8+P9 verschmelzen konzeptionell zu „Projekt & Pipelines". P8 baut den* ***`Projekt`-Kern*** *(typisierte Superset-Tabelle + CustomFields) + die* ***Pipeline-Engine*** *(frei konfigurierbare Views: Pipeline anlegen/benennen, Zugriff je Rolle, Feld-Sichtbarkeit + Pflichtfelder pro Pipeline/Status, Statuswerte) + konfigurierbaren* ***Kanban*** *(ein Board statt Sales-Kanban + Projekt-Kanban). „Deal"/„Auftrag" = Pipeline-Typen. Forecast/Volumen-Field-Level-Rechte bleiben als Pipeline-typ-bedingte Features. SC/Reqs werden mit dem ER-Modell neu gefasst.*

Goal: Ein Vertriebler führt Deals durch mehrere SHK-Pipelines mit Kanban, Forecast und feingranularer Sichtbarkeit von Volumen-Daten. **Depends on**: Phase 7 **Requirements**: REQ-sales-datenmodell, REQ-sales-kanban-listen-view, REQ-sales-deal-lifecycle, REQ-sales-forecast-kpis, REQ-sales-field-level-permissions **Source-Specs**: `_bmad-output/specs/M03-sales-pipeline.md` **Milestone-PLANs**: 7 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 8); inkl. M049 (Auto-Verkaufsfall, aus Phase 7 hierher) **Success Criteria** (what must be TRUE):

1. Mehrere Pipelines (Pipeline, PipelineStage, Deal, LostReason) mit fortlaufenden Deal-Nummern (HiveNumberSeq) existieren als geseedete SHK-Pipelines.

2. Ein Nutzer kann Deals im Kanban per Drag-&-Drop verschieben (mit Stage-Wechsel-Validation) oder in der Listen-View mit Filter/Bulk-Actions bearbeiten; Updates kommen in Real-Time (SSE, AD-9).
3. Ein qualifizierter Lead erzeugt automatisch einen Verkaufsfall (M049, konsumiert das `lead.qualified` -Event aus P7); ein gewonnener Deal löst den Won-Handoff an die Auftragspipeline (Phase 9) aus; ein Auto-Archive-Cron räumt abgeschlossene Deals.
4. Forecast, KPIs pro Vertriebler, Stage-Conversion-Rates und Lost-Reason-Auswertung werden mit Aggregat-Cache berechnet.
5. Field-Level-Permissions (deals.read_volume_own/all, read_aggregates) blenden Volumen-Werte für Nutzer ohne Recht aus. **Plans**: TBD **UI hint**: yes

Phase 9: Auftrags-/Projektpipeline → Projekt & Pipelines (Teil 2: Auftrags-Features + Sub-Areas)

> 🔴 ***Pivot-Note (AD-20, 13.06.):*** *Kein separates Project-Objekt mehr — die Auftrags-Welt ist eine* ***Pipeline-View auf denselben `Projekt` -Kern*** *(kein Won-Handoff, das Projekt wechselt nur die Pipeline/erhält Auftrags-Felder). P9 liefert die auftragsspezifischen Features:* ***Plantafel*** *(Scheduling pro Monteur),* ***Sub-Areas*** *am Projekt (Termine,* ***Rechnungen = neues Sub-Entity v1****, Checklisten, Aufgaben, Aktivitäten, Dokumente) und die* ***Abnahme als Checklisten/Protokoll-Sub-Area*** *(v1, rechtlich bindend, Touch-Signatur, offline; Mängel = Checkliste). Material/Zeiterfassung/Leistungspositionen/Baustelle/Support → v2-Backlog. SC/Reqs werden mit dem ER-Modell neu gefasst.*

Goal: Ein gewonnener Auftrag wird zum geplanten Projekt — mit Plantafel-Scheduling pro Monteur und digitaler, offline-fähiger Abnahme auf der Baustelle. **Depends on**: Phase 8 **Requirements**: REQ-projekt-datenmodell, REQ-projekt-plantafel, REQ-projekt-abnahme **Source-Specs**: `_bmad-output/specs/M04-auftragspipeline.md` **Milestone-PLANs**: 6 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 9); Scope-Schnitt 12.06.: M070 gedünnt — Lagerist-Sicht/Material-Ausgang → v2 (Material-Listen am Auftrag bleiben; Noah-Frage auf E-Liste) **Success Criteria** (what must be TRUE):

1. Projekte laufen über Multi-Pipelines (ProjectPipeline, Project, ProjectMember, AcceptanceProtocol, ProjectTemplate) und entstehen aus dem Sales-Won-Handoff.
2. Ein Disponent kann auf einer Plantafel Aufträge pro Monteur/Tag einplanen; die Ansicht ist offline-fähig.
3. Ein Monteur kann ein digitales Abnahme-Protokoll mit Unterschrift erfassen — auch offline, mit späterem Sync und M05-Form-Template-Unterstützung. **Plans**: TBD **UI hint**: yes

Phase 10: KWP-Integration — → v2 (KOMPLETT DEFERRED, 2026-06-12 — ERSTES v2-Modul)

> 🔴 ***Scope-Schnitt #12 (Kalender-Realität):*** *KWP-API-Zugang lt. Hersteller-Mail erst* ~~***19.07. — nach dem v1-Launch***~~ *(10.07.). KWP bleibt USP; v2 startet damit.* ***Sofort erledigt:*** *Callback-URL gemeldet (`https://app.hivegrid.de/api/apps/kwp/oauth/callback`, PKCE-Flow), bnWin.net-Update 8.1 N3 bei Geerkens angestoßen. v1 behält Apps-Engine + Extension-Points (Auftrag/Abnahme/Material). Beschreibung unten gilt für die v2-Planung.*

Goal: HiveGrid synchronisiert Auftrags- und Stammdaten bidirektional mit dem im SHK-Markt verbreiteten ERP KWP (Kernsystem des Erstkunden) — über die offizielle KWP-API als DataConnection, mit Schema-Drift-Schutz. Damit ist die Kette Lead → Sales → Auftrag → KWP durchgängig. **Depends on**: Phase 9, Phase 5 **Requirements**: REQ-kwp-datenmodell, REQ-kwp-architektur, REQ-kwp-schema-drift **Source-Specs**: `_bmad-output/specs/M19-kwp-integration.md` (⚠ M089–M095-Recut — siehe Phase-Note) **Milestone-PLANs**: 8 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 10); inkl. M074 (Task-ERP-Sync, aus Phase 4 hierher) **Success Criteria** (what must be TRUE):

1. KWP ist als `DataConnection type='rest-api'` in der Apps-Engine angebunden; keine KWP-DB-Credentials liegen in der Cloud (KwpSettings, KwpRegistry, KwpProjectQueue, KwpApiKey).
2. Auftrags-/Stammdaten fließen bidirektional zwischen HiveGrid und KWP; bei Konflikten gilt „KWP gewinnt" (AD-14/AD-17).
3. Ein Schema-Drift-Sentinel erkennt MSSQL-Schema-Änderungen per Snapshot-Vergleich (KwpSchemaSnapshot) und benachrichtigt den Tenant-Admin. **Plans**: TBD **Note**: M089-M095-Recut bindend — das offizielle KWP-API macht das Windows-Sync-Programm M089 obsolet (M090/091/092/094/095 vereinfachen sich, M093 bleibt). M089-M095 sind in der Plan-Phase neu zu schneiden, bevor Code entsteht; die alte kwpsync.exe-Drei-Komponenten-Architektur aus der M19-Spec ist NICHT zu bauen. KWP bleibt Backend-Sync ohne eigene UI (kein UI hint).

Phase 11: Globale Suche & Dashboard

Goal: Ein Nutzer findet jede Entität per Hotkey über das gesamte System und sieht seine Kennzahlen in einem rollenbasierten Widget-Dashboard. **Depends on**: Phase 9 **Requirements**: REQ-suche-datenmodell, REQ-suche-hybrid, REQ-suche-ui, REQ-dashboard-datenmodell, REQ-dashboard-widgets **Source-Specs**: `_bmad-output/specs/M24-globale-suche.md`, `_bmad-output/specs/M20-dashboard.md` **Milestone-PLANs**: 5 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 11); Scope-Schnitt 11.06.: Custom-Widgets (M149) + Export (M150) → v2; 12.06.: M147 gedünnt (feste Rollen-Layouts) **Success Criteria** (what must be TRUE):

1. Self-hosted Meilisearch liefert permission-aware Keyword-Treffer über den SearchAdapter. *(Scope-Schnitt 11.06.: pgvector/Hybrid-RRF → v2 mit HiveMind; AD-5 sequenziert.)*
2. Entity-Mutationen aktualisieren den Index über das Outbox-Pattern; SearchIndexEntry und RecentSearch sind befüllt.
3. Ein Nutzer öffnet per Cmd+K ein debounced Such-Modal mit RecentSearch und kann in eine Vollbild-Such-View wechseln.
4. Ein Nutzer sieht ein rollenbasiertes Dashboard (3 feste Rollen-Layouts GF/Büro/Monteur, SHK-Default geseedet) mit Widgets aus Sales, Projekte, Tasks, HR und KI. *(Scope-Schnitt 12.06.: Drag&Drop-Grid + User-Layout-Persistenz → v2; Widget-Registry bleibt voll — Inversion `M064→M147`.)*
5. Dashboard-Widgets aktualisieren sich live (WebSocket/SSE) mit Aggregat-Cache. **Plans**: TBD **UI hint**: yes

Phase 12: Automation-Engine

Goal: Die erste USP-Säule steht in voller Tiefe (AD-16): ein visueller Automation-Editor über alle Module mit vollständiger Trigger-/Action-Library, Guards und Sprach-zu-Automation-Erstellung. **Depends on**: Phase 11 **Requirements**: REQ-automation-datenmodell, REQ-automation-trigger-library, REQ-automation-action-library, REQ-automation-flow-editor, REQ-automation-schutz **Source-Specs**: `_bmad-output/specs/M17-automation-engine.md` **Milestone-PLANs**: 8 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 12) **Success Criteria** (what must be TRUE):

1. Ein Nutzer baut im Graph-Editor Flows aus der vollständigen Trigger-Library (CRM, Sales, Projekte, Email, Forms, Kalender, Zeit, Webhooks) und Action-Library (CRM, Notifications, HTTP, Tasks, If/Else, Delay, Loop; KWP-Push registriert sich mit v2-P10 nach — Registry ist erweiterbar gebaut).
2. Ein Nutzer erstellt eine Automation per natürlicher Sprache (DE), testet sie als Dry-Run und kann Ausführungen innerhalb von 24h zurückrollen.
3. Loop-Guard, Rate-Limit, Idempotenz-Key, Resume-Mode und Trigger-Volumen-Metering pro Tenant verhindern Runaway-Automationen.
4. AutomationRule (JSON-Graph), AutomationExecution, AutomationStepExecution und AutomationTemplate persistieren Flows und Läufe nachvollziehbar. **Plans**: TBD **UI hint**: yes

Phase 13: KI-Layer

Goal: Der EU-only KI-Unterbau steht produktiv: Multi-Provider-LLM-Gateway mit Kostenschutz und die Sprach-zu-Automation als exklusives Automation-Feature. *(Scope-Schnitt 11.06.: HiveMind/RAG, Whisper, AlephAlpha → v2 — v1 baut die Andock-Adapter.)* **Depends on**: Phase 12 **Requirements**: REQ-ki-llm-gateway **Source-Specs**: `_bmad-output/specs/M18-ki-layer.md` (HiveMind/RAG/Whisper-Teile → v2) **Milestone-PLANs**: 5 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 13) **Success Criteria** (what must be TRUE):

1. Das EU-LLM-Gateway (Mistral La Plateforme primär, Provider-Abstraktion — zweiter EU-Provider folgt v2; Infra-Wechsel 12.06.: Scaleway entfällt) trackt Tokens und bucht gegen das lokale AiCredits-Konto; harte Limits + Drossel greifen; keine US-Kern-Abhängigkeit.
2. Sprach-zu-Automation (M086) läuft EXKLUSIV im Automation-Kontext: Satz → visueller Flow-Vorschlag mit Rückfragen + Existenz-Prüfungen; aktiviert nie selbst.
3. Promptfoo-Suite (A.14) testet die ausgelieferten KI-Funktionen.
4. v2-Andock-Adapter existieren: `RetrievalProvider`-Interface, Outbox-Index-Events (via P11), Bestätigungs-Grenze als wiederverwendbares Muster. **Plans**: TBD **UI hint**: yes

Phase 14: Email & Marketing

Goal: Der Betrieb arbeitet seinen Email-Verkehr direkt in HiveGrid ab — bidirektionale Gmail/Outlook-Inbox mit CRM-Matching — und versendet DSGVO-konforme Marketing-Kampagnen. **Depends on**: Phase 6, Phase 5 **Requirements**:

REQ-email-datenmodell, REQ-marketing-datenmodell, REQ-marketing-double-optin, REQ-marketing-versand **Source-Specs**: `_bmad-output/specs/M09-email-inbox.md` (nur Sende-Domain/Inbound-Teile; OAuth-Sync → v2), `_bmad-output/specs/M10-email-marketing.md` **Milestone-PLANs**: 6 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 14); Scope-Schnitt 11.06.: M096–M098 (Postfach-Sync) → v2 **Success Criteria** (what must be TRUE):

1. Die Sende-Domain pro Mandant ist verifiziert (DKIM/SPF/DMARC, Auto-Probe); Mails aus HiveGrid (Automation, Marketing, manuell) gehen über Mailjet raus und erscheinen am Kontakt.
2. Eingehende Mails an die Tenant-Inbound-Adresse erzeugen Leads/Aktivitäten; die **BCC-Logging-Brücke** protokolliert in Gmail/Outlook geschriebene Mails am Kontakt (kein OAuth nötig).
3. Marketing-Kampagnen versenden DSGVO-konform (Double-Opt-In, Suppression-Check, One-Click-Unsubscribe) über BullMQ mit Rate-Limiting und Bounce-Handling.
4. Der Versand läuft über eine SMTP-Provider-Abstraktion (Mailjet — AD-6 gelockt 11.06.) ohne US-Kern-Abhängigkeit. **Plans**: TBD **UI hint**: yes

Phase 15: Telefonie & Team-Chat

Goal: Telefonie und interne Kommunikation laufen in HiveGrid — Click-to-Call/Inbound über AirCall mit Auto-Lead-Anlage und Real-Time-Team-Chat mit entity-linked Channels. **Depends on**: Phase 6, Phase 5 **Requirements**: REQ-telefonie-datenmodell, REQ-telefonie-provider-adapter, REQ-chat-datenmodell, REQ-chat-realtime, REQ-chat-features **Source-Specs**: `_bmad-output/specs/M12-telefonie.md` , `_bmad-output/specs/M13-team-chat.md` **Milestone-PLANs**: 9 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 15); Scope-Schnitt 11.06.: HalloPetra (M105) → v2 **Success Criteria** (what must be TRUE):

1. Click-to-Call und Inbound-Anrufe laufen über den AirCall-Adapter (Apps-Engine-Connection, verschlüsselte OAuth-Tokens); unbekannte Nummern legen automatisch einen Lead an; Anruf-Events stehen als Automation-Trigger bereit (verpasster Anruf → Rückruf-Aufgabe).
2. Team-Chat bietet Real-Time-Messaging (WebSocket, Presence, Typing, Read-Receipts, Redis-Pub/Sub) für drei Channel-Typen (public, private, entity-linked), AD-9.
3. Chat unterstützt @Mentions, File-Uploads (via Files), Link-Previews und Chat-Verlaufs-Suche.
4. Entity-linked Channels erscheinen im Deal-/Projekt-Detail und verknüpfen die Konversation mit der CRM-/Projekt-Entität. **Plans**: TBD **UI hint**: yes

Phase 16: Kalender, Booking, Forms & Signatur

Goal: Termin- und Dokument-Workflows sind abgedeckt — bidirektionaler Kalender-Sync, öffentliche Buchungsseiten, ein Drag-&-Drop-Form-Builder und Multi-Signer-Angebots-Signatur. **Depends on**: Phase 7, Phase 5 **Requirements**: REQ-kalender-datenmodell, REQ-kalender-sync-adapter, REQ-booking-datenmodell, REQ-booking-slot-kalkulierung, REQ-booking-integrationen, REQ-form-datenmodell, REQ-form-editor, REQ-form-actions, REQ-signatur-datenmodell, REQ-signatur-provider **Source-Specs**: `_bmad-output/specs/M25-calendar-sync.md` , `_bmad-output/specs/M06-booking.md` , `_bmad-output/specs/M05-form-builder.md` , `_bmad-output/specs/M07-angebot-signatur.md` **Milestone-PLANs**: 15 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 16); Scope-Schnitt 12.06.: M122 gedünnt — personalisierte Links + JS-Snippet → v2 (URL/QR/iFrame bleiben); M129 bleibt explizit voll **Success Criteria** (what must be TRUE):

1. Kalender synchronisiert bidirektional mit **Google** (CalendarAdapter, Auto-Token-Refresh, Webhook-Subscriptions) und legt HiveGrid-Termine an. *(Scope-Schnitt 11.06.: Outlook-Adapter → v2; Interface ist provider-agnostisch gebaut. Booking: eine Zuteilungs-Strategie „Erster der frei ist", Round-Robin/Wahl → v2.)*
2. Öffentliche Booking-Seiten berechnen Slots timezone-aware (Verfügbarkeitsregeln, Ausnahmen, Kalender-Sync, Puffer) und legen nach Buchung Lead, Activity und Bestätigungs-Email an (iFrame-Embed).
3. Ein Nutzer baut Formulare per Drag-&-Drop mit 12 Feld-Typen, Conditional Logic, SHK-Templates und iFrame-Embed; Post-Submit-Actions (Lead/Deal/Task, Webhook, Email, Automation-Trigger) feuern mit Field-Mapping und ActionLog.
4. Angebote durchlaufen Multi-Signer-Signaturen via PandaDoc/DocuSign über die Apps-Engine mit Webhook-Verarbeitung und Status-Updates. **Plans**: TBD **UI hint**: yes

Phase 17: HR-Hub

Goal: Interne Personalprozesse sind in einem verschlüsselten HR-Hub abgedeckt — Personalakten, Passwort-Safe und Urlaubs-/Krank-Workflow. **Depends on**: Phase 4 **Requirements**: REQ-hr-datenmodell, REQ-hr-passwort-safe, REQ-hr-

urlaub-krank **Source-Specs**: `_bmad-output/specs/M14-hr-hub.md` **Milestone-PLANs**: 7 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 17); Scope-Schnitt 11.06.: Memberspot/Boobora (M138) → v2; Surveys bleiben **Success Criteria** (what must be TRUE):

1. HR-Profile, Urlaubs-/Krank-/Payslip-/Request-Daten liegen AES-256-GCM-verschlüsselt vor; sensible Felder nutzen KMS-backed Envelope-Encryption (AD-8).
2. Jeder Mitarbeiter hat einen verschlüsselten Passwort-Safe mit Audit-Trail und DSGVO-Export.
3. Ein mehrstufiger Urlaubs-Approval-Workflow und Krankmeldungs-Erfassung speisen einen Abwesenheits-Kalender. **Plans**: TBD **UI hint**: yes

Phase 18: Customer-Portal — → v2 (deferred 2026-06-11)

Goal: *(v2)* Getrenntes Endkunden-Portal mit Magic-Link und freigegebener Projekt-Sicht. AD-18 revidiert — komplette Phase nach v2 verschoben; Phasen-Nummer bleibt reserviert. **Depends on**: — (v2-Planung) **Requirements**: *(→ v2: REQ-portal-datenmodell, REQ-portal-auth-magic-link, REQ-portal-projekt-sicht)* **Source-Specs**: `_bmad-output/specs/M15-customer-portal.md` **Milestone-PLANs**: 0 in v1 (M141–M146 → v2) **v1-Andockpunkte (bleiben):** P4-Portal-Upload-Vorbereitung (FileEntityLink), `portal.hivegrid.de` im P1-URL-Schema. **Plans**: — (v2)

Phase 19: Plattform-Admin

Goal: Die Plattform ist aus einer getrennten Admin-Welt betreibbar — Tenant-Management, Subscription-Overrides, Feature-Flag-Verwaltung (UI zur Engine aus Phase 1), Queue-Monitoring und Support-Inbox. **Depends on**: Phase 2 **Requirements**: REQ-plattform-admin-identität, REQ-plattform-admin-features **Source-Specs**: `_bmad-output/specs/M37-plattform-admin.md` , `_bmad-output/specs/M32-feature-flags.md` (Admin-UI-Teile) **Milestone-PLANs**: 7 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 19); Scope-Schnitt 11.06.: M033 (Flag-Massen-Statistik) + M040 (Plattform-Lead-Pipeline) → v2; 12.06.: M038/M039 (getrennte AdminUser-Identität) → v2, M035 gedünnt. **Neu:** Admin stellt Tenant-Abos zusammen (Konnektoren schalten + Stripe-Positionen — Abo-Baukasten-Mechanik; Self-Service-UI → v2) **Success Criteria** (what must be TRUE):

1. Ein Plattform-Admin meldet sich auf `admin.hivegrid.de` an (Host-Routing auf die Admin-Routen, **Cookie-Domain-Trennung** zu app.hivegrid.de, TOTP-Pflicht + MFA-Step-up); Admin-User haben **keinen Tenant-Workspace** und erscheinen in keiner Mandanten-Mitgliederliste; kein Generalschlüssel zwischen Plattform und Mandanten. *(Scope-Schnitt 12.06.: getrennte AdminUser-Identität/AdminRBAC-Parallelwelt = M038/M039 → v2; v1 nutzt Boilerplate-Modell User-Extension + `admin-mfa-gate` .)*
2. Ein Admin verwaltet Tenants, Subscription-Overrides und Feature-Flags aus einer zentralen Konsole; das Admin-Dashboard zeigt die Boilerplate-KPIs (Tenants/User/MRR). *(Churn/ARPU/Health-Kacheln → v2.)*
3. Ein Admin überwacht BullMQ-Queues und bearbeitet eine Support-Inbox; AdminAuditLog protokolliert plattformweite Aktionen (tenantId=null) über die Audit-Hash-Kette. **Plans**: TBD **UI hint**: yes

Phase 20: Billing, Onboarding & Branding

Goal: Ein neuer Betrieb kommt vom Self-Service-Signup per Auto-SHK-Provisionierung ins Produkt (Wizard-UI → v2), läuft auf Stripe-Billing und kann sein Theme wählen. **Depends on**: Phase 19, Phase 6 **Requirements**: REQ-billing-datenmodell, REQ-billing-trial-dunning, REQ-onboarding-demo-daten, REQ-branding-theme-phase1 **Source-Specs**: `_bmad-output/specs/M28-subscriptions-billing.md` (⚠ M154-Recut: Abo-Baukasten statt Seat-Typen), `_bmad-output/specs/M29-onboarding.md` (Wizard-UI → v2), `_bmad-output/specs/M31-branding.md` ; **Legacy-Referenz:** `archive/hivegrid-cloud/app/config/pricing-plans.ts` **Milestone-PLANs**: 8 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 20); Scope-Schnitt 11.06.: M157 (Wizard-UI) → v2; 12.06.: M030 + M160 gedünnt **Success Criteria** (what must be TRUE):

1. Billing läuft über Stripe als **Abo-Baukasten**: Basis-Preis pro Nutzer + Feature-Module als daten-getriebene Positionen (keine Seat-Typen — M154-Recut); Trial mit Dunning (Tag 1/3/7) und Frozen-Modus (Read-Only-Middleware) greift; konkrete Preise geparkt (AD-15, Kostenmatrix folgt). Verbrauchs-Zähler + Drossel aktiv, Aufschlags-Abrechnung → v2.
2. Ein neuer Tenant wird nach Signup automatisch SHK-provisioniert (Seeds M158, ohne Wizard-UI — die folgt v2); Trial-Freischaltung durch Plattform-Admin (wie spec'd).
3. Optionale Demo-Daten (5 Contacts, 3 Deals, 2 Projekte, Tasks) lassen sich per Settings-Knopf provisionieren und via is_demo_data-Flag löschen. *(Scope-Schnitt 12.06.: KI-Walkthrough → v2 mit Wizard; einfaches Löschen statt Multi-Step-Bulk.)*

4. Ein Nutzer schaltet zwischen Hell und Dunkel um, flash-frei (SSR-Cookie). *(Scope-Schnitt 12.06.: kein System-Modus — bewusst nur 2 Optionen;* `next-themes` *aus Boilerplate.)* **Plans**: TBD **UI hint**: yes

Phase 21: DSGVO & Launch-Reife

Goal: Der DSGVO-Lebenszyklus ist abgedeckt und die Plattform launch-fertig. Damit ist der Big-Bang-Launch (AD-19) auslösbar. *(Scope-Schnitt 11.06.: Public-API, Mobile-PWA-Ausbau, Help-Center/Desk, DSGVO-Auskunfts-Tooling → v2.)* **Depends on**: Phase 20, Phase 10, Phase 4 **Requirements**: REQ-dsgvo-export, REQ-dsgvo-tenant-deletion, REQ-feedback-widget **Source-Specs**: `_bmad-output/specs/M35-dsgvo-export.md` (M34/M36/M38/M39 → v2) **Milestone-PLANs**: 2 PLANs in `docs/milestones/` → siehe `.planning/PHASE-SOURCES.md` (Phase 21) **Success Criteria** (what must be TRUE):

1. Ein Nutzer kann einen DSGVO-ZIP-Export (72h-signed-URL) anstoßen; eine Tenant-Löschung führt Crypto-Shredding (Per-Tenant-DEK), Stripe-Cancel und Storage/Cache-Bereinigung durch (AD-8).
2. DSGVO-Auskunft funktioniert als dokumentiertes manuelles Playbook: Suche → Kontakt-Detail (Activity-Stream) → CSV-Export → Anwalts-Vorlage → Frist-Task (+25 Tage). AVV-Vorlage liegt vor (Externals E6).
3. Das In-App-Feedback-Widget (Bug/Feature/Frage, Screenshot, Rate-Limit) ist live und speist die Admin-Support-Inbox (P19, generische Quelle); eine Minimal-FAQ-Seite existiert.
4. Monteure arbeiten mobil über die PWA/responsive Web-App (mobile-first Screens + P9-Offline) — der PWA-Ausbau (Installierbarkeit/Web-Push) folgt v2; ein nativer Client ist nicht vorgesehen (AD-17, PWA-only). **Plans**: TBD **UI hint**: yes

---

v2 — Phase Details (22–34)

> *Detail-Quelle mit allen Success-Criteria/Screens/Recut-Hinweisen:* `.planning/ROADMAP-V2.md` *. Dependency-Prinzip wie v1: keine Vorwärts-Abhängigkeiten; spätere Phasen integrieren sich rückwirkend.*

Phase 22: Public-API & Webhooks (v2)

Goal: Externe Systeme lesen HiveGrid-Hauptentitäten per versionierter REST-API und empfangen/senden Events über signierte Webhooks. **Depends on**: Phase 2 (ApiKey-Modell), Phase 12 (Automation `external_event` -Trigger) **Milestones**: M161, M162, M163, M027 **Source-Specs**: `_bmad-output/specs/M34-public-api.md` **Success Criteria** (what must be TRUE):

1. GET contacts/companies/deals/projects/tasks/leads — cursor-paginiert, Filter/Sort/Sparse-Fieldsets/Embed (Tiefe 1); `read_only` -Scope lässt sensible Felder aus (nicht 403).
2. Eingehende Lead-Webhooks legen Leads an (API-Key/HMAC/URL-Token + Payload-Mapping + Auto-Match); ausgehende Subscriptions HMAC-signiert mit Idempotenz, Retry/Backoff, Auto-Disable nach 10 Fehlern.
3. OpenAPI-3.1 auto-generiert + Swagger-UI; Rate-Limit (1k/h Key, 10k/h Tenant) mit `X-RateLimit-*` + Klartext-429. **Plans**: TBD **UI hint**: yes

Phase 23: Postfach-Sync (Email-Inbox) (v2)

Goal: Gmail/Outlook-Postfächer werden bidirektional ins CRM gespiegelt, auto-gematcht und im Activity-Stream sichtbar. **Depends on**: Phase 5 (Apps-Engine/Provider-Pattern), Phase 6 (CRM-Matching), Phase 14 (Sende-Domain — nicht neu bauen) **Milestones**: M096, M097, M098 **Source-Specs**: `_bmad-output/specs/M09-email-inbox.md` (OAuth-Sync/Inbox/Threading-Teile) **Success Criteria** (what must be TRUE):

1. OAuth-Connect Gmail (Pub/Sub-Watch) + Microsoft (Graph-Subscription, 3-Tage-Renewal-Worker); Push-Sync, Initial-Sync 90 Tage resumebar.
2. Auto-Match (From-Adresse, Subject-Pattern, 1-offener-Deal-Heuristik) mit manuellem Override; Inbox-Liste + Detail mit Thread, sicherem HTML-Render, Reply/Forward.
3. `email.read_own` / `read_all` filtern Liste + Activity-Stream. **Plans**: TBD **UI hint**: yes **Risiko**: Gmail restricted scopes → Google-Verifizierung/CASA (Antrag lief in v1 EXT-13); Graph-Subscription-Renewal kritisch.

Phase 24: Self-Service-Billing & Onboarding-Wizard (v2)

Goal: Tenants buchen/wechseln Pläne self-service über Web-Checkout und durchlaufen einen branchen-getriebenen Onboarding-Wizard. **Depends on**: Phase 20 (Stripe/Billing-Fundament) **Retro-Integration**: KI-Walkthrough-Coach wird von Phase 25 rückwirkend in den Wizard eingehängt (kein harter Forward-Dep). **Milestones**: M157 (+ M28-Self-Service-

Delta, Block P) **Source-Specs**: `_bmad-output/specs/M28-subscriptions-billing.md` (Block P), `M29-onboarding.md` **Success Criteria** (what must be TRUE):

1. Self-Service-Trial ohne Vertriebs-Freigabe (Zahlungsmethode-Pflicht); Plan-Wechsel/Add-Ons über Web-Checkout (PWA-only → kein App-Store, keine Store-Billing-Auflage).
2. Onboarding-Wizard resumebar, Step-Validierung, Rückwärts-Block nach irreversiblen Actions.
3. SHK-Branchen-Template (Pipelines/Rollen/Felder/Forms/Demo/Checkliste) deklarativ geseedet; Erste-Schritte-Checkliste + Demo-Daten-Bulk-Löschung. **Plans**: TBD **UI hint**: yes **Risiko**: Apple Guideline 3.1.3(c) ist der Grund, warum Self-Service v2 ist; konkrete Preise (v1 EXT-14) Voraussetzung.

Phase 25: KI-Tiefe — HiveMind/RAG (v2)

Goal: Aus dem v1-LLM-Gateway wird semantische Tiefe — RAG über Tenant-Daten, EU-Whisper, perspektivisch der autonome HiveMind-Agent. **Depends on**: Phase 13 (LLM-Gateway), Phase 11 (Outbox/Suche für Hybrid-RRF) **Retro-Integration**: Phase 29 (Help) registriert seine Artikel rückwirkend als RAG-Korpus (Phase 25 liefert RetrievalProvider + Embedding-Pipeline). **Milestones**: M087 (+ Whisper-EU, pgvector-Hybrid-Suche, zweiter EU-Provider) **Source-Specs**: `_bmad-output/specs/M18-ki-layer.md` (RAG/HiveMind/Whisper-Teile) **Success Criteria** (what must be TRUE):

1. Embedding-Pipeline → `TenantEmbedding` (pgvector), Re-Embed per Hash-Diff, Permission-Pre-Filter vor Vector-Search; hybride globale Suche (Volltext+Vektor, RRF).
2. Whisper-EU transkribiert Voice-Notes async; Multi-Provider-Failover (Mistral→Scaleway→AlephAlpha) + Promptfoo in CI.
3. HiveMind-Agent liefert proaktive Vorschläge mit Action-Boundary (keine schreibenden Aktionen ohne Bestätigung), Audit-Trail, Scheduled-Actions. **Plans**: TBD **UI hint**: yes **Constraint**: ausschließlich EU-Provider (DSGVO), AVV mit allen Providern.

Phase 26: Customer-Portal (v2) ⭐ *(ex-P18)*

Goal: Endkunden sehen über eine eigene Auth-Schicht ihre Projekte/Status/Dokumente, laden Dateien hoch, erhalten Auto-Status-Mails. **Kritisch** — wenn der Sprint im Plan liegt, in v2 liefern. **Depends on**: Phase 4 (FileEntityLink — in v1 vorbereitet), Phase 6 (CRM/Project), Phase 9 (Project-Stages/Acceptance), Phase 14 (Sende-Domain) **Milestones**: M141, M142, M143, M144, M145, M146 **Source-Specs**: `_bmad-output/specs/M15-customer-portal.md` **Success Criteria** (what must be TRUE):

1. Eigene `PortalUser`-Klasse strikt getrennt von TenantUser (eigene Cookies/Routes, Magic-Link primär); kein Permission-Leak, fehlender Zugang → 404.
2. Endkunde sieht nur eigene Projekte/konfigurierte Stages/freigegebene Dokumente; Upload (Kamera, Multi-File) → Files + Activity + PL-Notification.
3. Auto-Status-Mails bei `project.stage_changed` (Debounce/Bedingungen); Tenant-Admin konfiguriert Sichtbarkeit/Templates; DSGVO-Lösch/Export für PortalUser. **Plans**: TBD **UI hint**: yes (extern sichtbar, eigener Polish)

Phase 27: Mobile-PWA-Ausbau (v2)

Goal: Die mobile-first-PWA aus v1 wird zur vollwertigen installierbaren App — Add-to-Homescreen, Service-Worker-Offline-Härtung, Web-Push, schneller Baustellen-Einsatz. **Kein nativer Client** (Expo/Capacitor) und **kein App-Store** — native → frühestens v3 (AD-17, PWA-only). **Depends on**: Phase 9 (OfflineSyncQueue), Phase 21 (notify()-Push-Kanal → Web-Push), Phase 2 (Auth/Magic-Link) **Milestones**: M167, M168, M169, M170, M171 (auf PWA umzuschneiden — kein EAS-Build, keine Store-Compliance) **Source-Specs**: `_bmad-output/specs/M36-mobile-app.md` (⚠ Recut → PWA-only; Capacitor/Expo verworfen) **Success Criteria** (what must be TRUE):

1. Offline: Pre-Cache beim Login + Offline-Write-Queue (Foto/Note/Task-Status/Checklist/Acceptance/Chat) über Service-Worker + IndexedDB; Sync-on-Reconnect idempotent (lokale UUID), Konflikt-Resolution.
2. Installierbar: Web-App-Manifest + Add-to-Homescreen + Standalone-Display; Kamera/Foto via Web-APIs mit lokalem Resize.
3. Web-Push (VAPID) aktiviert den in P1 ruhenden notify()-Push-Kanal; Account-Löschung erreichbar; Force-Update via Service-Worker-Versionierung. **Plans**: TBD **UI hint**: yes (mobile-first PWA-Shell, Touch-Paradigma) **Risiko**: PWA-Offline-Storage-Budget + iOS-Safari-PWA-Grenzen (Web-Push iOS ≥16.4 nur im Standalone-Modus); Background-Sync-Verfügbarkeit. **Kein** App-Store-Review-Risiko (entfällt vollständig).

Phase 28: Plattform-Admin-Tiefe (v2)

Goal: Getrennte AdminUser-Identität/RBAC auf admin.hivegrid.de + Feature-Flag-Massen-Statistik (löst die v1-Auflagen-Brücke ab). **Depends on**: Phase 19 (Admin-Welt-Fundament/M034: Host-Routing/Cookie-Trennung/TOTP), Phase 2 (Auth-Bausteine) **Milestones**: M038, M039, M033, M040 **Source-Specs**: `_bmad-output/specs/M37-plattform-admin.md` (Block A–D), `M32-feature-flags.md` (Massen-Statistik) **Success Criteria** (what must be TRUE):

1. Eigene `AdminUser`-Klasse hart getrennt (eigene Cookies SameSite=Strict, **kein BYPASSRLS** — Admin liest Tenant-Daten via RLS mit gesetztem `app.tenant_id`); AdminRBAC + 2FA-Pflicht + Self-Permission-Schutz.
2. AdminAuditLog cross-tenant mit Reason-Pflicht; Impersonate mit Reason + Banner.
3. Flag-Massen-Statistik (Effektiv-Auswertung + Live-Filter-Vorschau + Bulk-Override), deterministische Auswertung (Override > Filter > Default); interne Lead-Pipeline. **Plans**: TBD **UI hint**: yes

Phase 29: Help-Center & Help-Desk (v2)

Goal: Öffentliches file-basiertes Help-Center mit Deep-Links aus der App + In-App-Feedback-Widget (Ausbau), das im Plattform-Admin landet. **Depends on**: Phase 21 (Feedback-Widget→Support-Inbox-Andockpunkt), Phase 19 (Admin) **Retro-Integration**: liefert HelpArticle-Embeddings als RAG-Korpus an Phase 25. **Milestones**: M172, M173, M174, M175 **Source-Specs**: `_bmad-output/specs/M38-help-center.md`, `M39-help-desk.md` **Success Criteria** (what must be TRUE):

1. Markdown mit Frontmatter, Build-Time-Index (Volltext/Kategorien/Sitemap), Build-Fail bei kaputten Referenzen; öffentliche Routen `/help/{lang}/{category}/{slug}` ohne Auth + SEO + Suche.
2. Deep-Link Route→Artikel (Hilfe-Icon pro App-Route); Feedback-Widget → `Feedback` mit Auto-Kontext + Screenshot + Rate-Limit.
3. Plattform-Admin: Feedback-Liste + Detail (Status-Lifecycle/Assignee/Notes) mit Audit. **Plans**: TBD **UI hint**: yes

Phase 30: Dashboard-Deluxe & Querschnitt (v2)

Goal: Querschnitts-Komfort — anpassbare Dashboards/Export, Notification-Bündelung/Quiet-Hours, Sprach-Wechsler/EN. **Depends on**: Phase 11 (Dashboard/Widget-Registry), Phase 3 (Notification-Center), Phase 1 (i18n-Library) **Milestones**: M149, M150, M117, M029 **Source-Specs**: `_bmad-output/specs/M20-dashboard.md`, `M22-notifications.md`, `M30-i18n.md` **Success Criteria** (what must be TRUE):

1. Dashboard: Layout anpassbar (Drag/Resize, permission-gegateter Widget-Catalog), Drilldown, CSV-Export, Realtime mit Opt-Out.
2. Notifications: Bündelung (5-Min-Window → 1 Push/Mail mit count), Quiet-Hours (queued, Urgent-Override), Pro-User-Präferenzen Event×Kanal.
3. i18n: Sprach-Wechsler + EN-Locale (DeepL DE→EN mit Review), Build-Validierung, Fallback DE, Empfänger-Sprach-Auflösung Outbound. **Plans**: TBD **UI hint**: yes

Phase 31: DSGVO-Vollausbau (v2)

Goal: DSGVO vertiefen — AVV-Versionierung, Subprozessoren-Workflow, Auskunfts-/Voll-Export-Workflow (ersetzt das manuelle v1-Playbook). **Depends on**: Phase 21 (DSGVO-Export/Lösch-Fundament), Phase 28 (Admin-Legal-Verwaltung/AdminAuditLog) **Milestones**: M166 **Source-Specs**: `_bmad-output/specs/M35-dsgvo-export.md` **Success Criteria** (what must be TRUE):

1. AVV-Versionierung mit dynamischem PDF + elektronischer Akzeptanz, Re-Akzeptanz bei Updates (14-Tage-Frist → Read-Only).
2. Subprozessoren-Liste mit Drittland-Warnung + 14-Tage-Widerspruchs-Workflow.
3. User-Daten-Export (Art. 15/20) + Tenant-Voll-Export (ZIP, async, 7-Tage-Window) + Tenant-Lösch-Workflow (30-Tage-Grace + Hard-Delete-Cron). **Plans**: TBD **UI hint**: yes **Risiko**: AVV-PDF-Inhalt anwaltlich finalisieren (v1 EXT-03 beauftragt).

Phase 32: KWP-Integration (v2 — RECUT) *(ex-P10)*

Goal: KWP wird regulärer REST/OAuth-Adapter über die Apps-Engine — bidirektionaler Sync (Aufträge/Stammdaten/Belege/Plantafel/Dokumente) mit „KWP gewinnt"-Konfliktregel. **Depends on**: Phase 5 (M33a Apps-Engine-Adapter-Interface — HART), Phase 9 (Auftrag/Abnahme/Material-Extension-Points), Phase 12 (Automation `kwp.push`-Action) **Trigger (Block 2)**: KWP-API-Zugang ~20.07.; agil vorziehbar, sobald Vertrag/Zugang da. **Milestones**:

M089, M090, M091, M092, M093, M094, M095, M074 **Source-Specs**: `_bmad-output/specs/M19-kwp-integration.md` + **`docs/milestones/M19-KWP-API-RECUT.md` (bindend) Success Criteria** (what must be TRUE):

1. KWP-API-Connector als erster konkreter Adapter hinterm M33a-Interface; Credentials per-Tenant-DEK in `DataConnection.config`.
2. Bidirektionaler Sync (Push via Outbox+Replay, Pull via Webhook/Cron, idempotent über `AppSyncItem`); Konflikt „KWP gewinnt" (LWW + Audit-Restore) + Optimistic Locking.
3. API-Contract-Sentinel (zod-Validierung + Versions-Check) pausiert Sync bei Drift + alarmiert Admin; Plantafel- + Dokumenten-Archiv-Sync über die API. **Plans**: TBD **UI hint**: yes **Note**: ⚠ kwpsync.exe NICHT bauen (3→2-Komponenten). Konkreter KWP-API-Vertrag (Auth/Endpoints/Webhook-Support/Doc-API) ist der Bau-Blocker.

Phase 33: Telefonie-Erweiterung (HalloPetra) (v2)

Goal: HalloPetra als zweiter Telefonie-Provider hinterm vorhandenen `TelephonyProviderInterface` — ohne Kern-Umbau. **Depends on**: Phase 15 (TelephonyProviderInterface + AirCall-Adapter als Referenz) **Trigger (Block 2)**: HalloPetra-API-Vertrag/Spec; agil vorziehbar, sobald spezifiziert. **Milestones**: M105 **Source-Specs**: `_bmad-output/specs/M12-telefonie.md` (Provider-Pattern) — ⚠ **HalloPetra selbst ist in keiner Spec; Vorlauf-Spec nötig. Success Criteria** (what must be TRUE):

1. HalloPetra-Adapter implementiert `TelephonyProviderInterface` (connect/triggerOutboundCall/subscribeToWebhooks/verifyWebhookSignature/parseWebhookEvent/getRecordingUrl) ohne Kern-Änderung.
2. Webhook-Auto-Match auf Contact/Lead/Deal, Activity + Automation-Trigger; Click-to-Call + Inbound providerunabhängig. **Plans**: TBD **UI hint**: yes (Setup-/Binding-Flächen)

Phase 34: HR-Drittanbieter-Onboarding (v2)

Goal: Externe HR-Onboarding-/Lohn-Drittanbieter (Memberspot/Boobora) über die Apps-Engine anbinden. **Depends on**: Phase 5 (Apps-Engine), Phase 17 (HR-Hub) **Trigger (Block 2)**: Memberspot/Boobora-API-Vertrag/Spec; agil vorziehbar, sobald spezifiziert. **Milestones**: M138 **Source-Specs**: `_bmad-output/specs/M14-hr-hub.md` — ⚠ **Memberspot/Boobora in keiner Spec; Vorlauf-Spec nötig. Success Criteria** (what must be TRUE):

1. HR-Onboarding-Checklisten pro neuem MA (verlinkte Forms/Tasks/Reminders).
2. Drittanbieter (Memberspot/Boobora/Lohn) als DataConnection über die Apps-Engine angebunden (gleicher Adapter-Mechanismus wie KWP). **Plans**: TBD **UI hint**: yes

Progress

Execution Order: Phases execute in numeric order: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 **v1-Launch nach Phase 21 (~10.07.).** Danach **v2 (Phasen 22–34, ~13.–31.07.):** Block 1 (22→31) in 2 dependency-freien Tracks; Block 2 (32 KWP, 33 HalloPetra, 34 HR-Drittanbieter) ans Ende, agil vorziehbar sobald Partner-API/Spec vorliegt. P10/P18 sind als v1-Nummern reserviert und werden in v2 als Phase 32 (KWP) bzw. Phase 26 (Portal) ausgeführt.

Phase	Plans Complete	Status	Completed
1. Fundament & Walking-Skeleton	0/TBD	Not started	-
2. Identität & Mandantentrennung	0/TBD	Not started	-
3. Querschnitts-Oberflächen	0/TBD	Not started	-
4. Files & Tasks	0/TBD	Not started	-
5. Apps-Engine & Integrations-Framework	0/TBD	Not started	-
6. CRM-Kern	0/TBD	Not started	-
7. Lead-Inbox	0/TBD	Not started	-
8. Sales-Pipeline	0/TBD	Not started	-

9. Auftrags-/Projektpipeline	0/TBD	Not started	-
10. KWP-Integration	0/TBD	Not started	-
11. Globale Suche & Dashboard	0/TBD	Not started	-
12. Automation-Engine	0/TBD	Not started	-
13. KI-Layer	0/TBD	Not started	-
14. Email & Marketing	0/TBD	Not started	-
15. Telefonie & Team-Chat	0/TBD	Not started	-
16. Kalender, Booking, Forms & Signatur	0/TBD	Not started	-
17. HR-Hub	0/TBD	Not started	-
18. Customer-Portal	→ v2	Deferred (11.06.)	-
19. Plattform-Admin	0/TBD	Not started	-
20. Billing, Onboarding & Branding	0/TBD	Not started	-
21. DSGVO & Launch-Reife	0/TBD	Not started	-
— v2 (Sprint 13.–31.07.) —			
22. Public-API & Webhooks	0/TBD	Not started (v2)	-
23. Postfach-Sync (Email-Inbox)	0/TBD	Not started (v2)	-
24. Self-Service-Billing & Onboarding-Wizard	0/TBD	Not started (v2)	-
25. KI-Tiefe — HiveMind/RAG	0/TBD	Not started (v2)	-
26. Customer-Portal ⭐	0/TBD	Not started (v2, ex-P18)	-
27. Mobile-PWA-Ausbau	0/TBD	Not started (v2)	-
28. Plattform-Admin-Tiefe	0/TBD	Not started (v2)	-
29. Help-Center & Help-Desk	0/TBD	Not started (v2)	-
30. Dashboard-Deluxe & Querschnitt	0/TBD	Not started (v2)	-
31. DSGVO-Vollausbau	0/TBD	Not started (v2)	-
32. KWP-Integration (Recut)	0/TBD	Block 2 — API-Zugang ~20.07.	-
33. Telefonie-Erweiterung (HalloPetra)	0/TBD	Block 2 — Spec offen	-
34. HR-Drittanbieter-Onboarding	0/TBD	Block 2 — Spec offen	-

---

*Roadmap erzeugt: 2026-06-07 via /gsd:new-project (Stufe 2 — Phasenstruktur). Revidiert: 2026-06-07 — 14 → 21 Phasen (dicke Buckets aufgeteilt: Automation/KI, Kommunikation, HR/Portal, Plattform/Launch; Querschnittsdienste in Plumbing + Files/Tasks). 109/109 v1-Requirements gemappt. Bau-Reihenfolge nach AD-11; Go-Live Big-Bang nach AD-19. Revidiert: 2026-06-10 nach Dependency-Audit (61 Vorwärts-Verletzungen geprüft) — Service-Kerne M001/M003/M006/M011 von P3 → P1 (P3 umbenannt zu „Querschnitts-Oberflächen"), M074 → P10, M031–M033 → P19; Deps von P4/P5/P19 auf Phase 2 gelockert. Revidiert: 2026-06-10 (zweiter Pass) — Dependency-Inversion vollständig: M049 → P8 (einziger echter Entity-Move); alle 40 verbleibenden Vorwärts-Refs per Retro-Integrations-Ownership invertiert (Register + Extension-Point-Pflichten + Audit-Kontrakt in PHASE-SOURCES.md). 0 unaufgelöste Vorwärts-Abhängigkeiten. Revidiert: 2026-06-11/12 — **Scope-Schnitt v1/v2** (10 Entscheidungs-Blöcke, Historie in SCOPE-V1-CUTS.md): v1 = 146 MS / 94 Reqs; 30 MS → v2 (Mobile→PWA-Ausbau, Portal, Postfach-Sync, Public-API, Help,*

HiveMind/Whisper, HalloPetra, Wizard-UI, Dashboard-Deluxe, Admin-Teile, M166). P18 deferred, P21 umbenannt. Neu: Entitlement-Gate-Pflicht (Konnektor je Modul, P1-Contract) + M154-Recut „Abo-Baukasten" (Legacy-Referenz pricing-plans.ts: Core/Swarm/Colony). AD-5 sequenziert, AD-17/AD-18 revidiert. Revidiert: 2026-06-12 — ***Scope-Schnitt #11 (Boilerplate-Recheck):*** *v1 = 141 MS / 93 Reqs; M027 (API-Keys), M029 (Sprach-Wechsler — v1 DE-only), M038/M039 (getrennte AdminUser-Identität — v1 = Boilerplate-Modell mit Auflagen: kein Tenant-Workspace, admin.hivegrid.de-Host-Routing + Cookie-Trennung, TOTP-Pflicht), M117 (Bündelung/Quiet-Hours) → v2. Dünnungen: M028, M035, M054 (HEIC→JPG bleibt Pflicht), M055, M147 (feste Rollen-Layouts), M152. Stale-Fixes: P11/P13/P14-Übersichtszeilen, URL-Schema Portal = Pfad gemäß INTEGRATION-CONTRACTS §6. Revidiert: 2026-06-12 (dritter Pass) —* ***Scope-Schnitt #12:*** *v1 =* ***133 MS / 90 Reqs****, v2 = 43 MS.* ***P10 KWP komplett → v2 als ERSTES v2-Modul*** *(Kalender-Realität: API-Zugang ~19.07. > Launch ~10.07.; Callback-URL gemeldet, bnWin.net 8.1 N3 angestoßen, Andockpunkte bleiben). Dünnungen: M030 (nur Hell/Dunkel), M070 (Lagerist → v2), M122 (personalisierte Links/JS-Snippet → v2), M160 (KI-Walkthrough → v2). M129 bleibt explizit voll. Inversions-Register 31→28 (KWP-Inversionen wandern mit v2-Owner). Ergänzt: 2026-06-15 —* ***v2-Phasen ausgebaut (Phase 22–34, GSD-Format).*** *Die 43 v2-Milestones thematisch zu 13 Phasen geschnitten (Coverage 43/43), Detail-Fassung in* `.planning/ROADMAP-V2.md` *. Block 1 (22–31) sofort baubar; Block 2 (32 KWP / 33 HalloPetra / 34 HR-Drittanbieter) ans Ende — „Partner-API-Vertrag offen", agil vorziehbar. Dependencies geprüft: keine Vorwärts-Abhängigkeiten (KI-Coach→Wizard und Help→RAG-Korpus per Retro-Integration aufgelöst). Bei den „phase:1"-Modulen (26/28/29/30) ist nur das v2-Delta verplant, nicht das v1-Fundament. HalloPetra (M105) + Memberspot/Boobora (M138) sind in keiner Spec → Vorlauf-Spec nötig.*