HiveGrid 2.0 — Phasen-Report

Bau-Roadmap für das Team-Review · Stand: 12.06.2026 (inkl. Scope-Schnitt v1/v2 vom 11.06. + Boilerplate-Recheck-Schnitt #11 vom 12.06.) Quellen: `.planning/ROADMAP.md` · `.planning/PHASE-SOURCES.md` · `.planning/REQUIREMENTS.md` · `docs/ARCHITECTURE-DECISIONS.md` (AD-1…AD-19) · 36 Modul-Specs · 176 Milestone-PLANs

---

1. Überblick

HiveGrid 2.0 ist eine Multi-Tenant-SaaS-Plattform für SHK-Handwerksbetriebe. Multi-Kunden-SaaS für den gesamten SHK-Markt; die **Geerkens GmbH** (Kontakt: Noah) ist erster Kunde und Investor. Das Produkt ersetzt HubSpot, Quiply, Zapier/n8n und mehrere Excel-Prozesse. Kernwert: Die durchgängige Kette **Lead → Sales → Auftrag → KWP-Sync** inkl. visueller KI-Automation in einem DSGVO-konformen EU-System.

Eckdaten der Roadmap:

Kennzahl	Wert
Phasen	**21** (Bau-Reihenfolge numerisch 1 → 21)
v1-Requirements	**90** (Scope-Schnitte 11.+12.06.: 19 → v2; lückenlos gemappt, 0 Orphans)
Detail-Milestones (BMAD)	**133 in v1 + 43 committed in v2** = 176 PLAN.md unter `docs/milestones/`
Modul-Specs	**36** unter `_bmad-output/specs/`
Go-Live	**Big-Bang** (AD-19): erst live, wenn das volle Phase-1-Produkt steht. Gegenmaßnahme: Noah validiert pro Modul über den PROGRESS-TRACKER-Klick-Pfad.
Vorwärts-Abhängigkeiten	**0 unaufgelöste** (Dependency-Audit + Inversion, 10.06.2026)

Die vier Bau-Prinzipien (gelten in jeder Phase)

1. **Hot-Swap-UI (Phase 1 baut die Maschine):** Jede Route liefert ein **View-Model** (reine Daten, kein React); UI-Komponenten erhalten nur VM + Action-Callbacks; **Subframe**-Designs werden über dünne **Adapter** gebunden. CI erzwingt das: Smoke-Test pro Route gegen `ui-mock`, Visual-Regression-Gate, Hot-Swap-Drill. → Designer-Screens sind jederzeit nachrüstbar, Logik bleibt unberührt.
2. **Dependency-Inversion:** Spätere Module integrieren sich **rückwirkend** in frühere Oberflächen. Frühe Phasen liefern Extension-Points (Events, Stubs, VM-Slots); die spätere Phase besitzt die Integrations-Aufgabe. Ownership-Register in `PHASE-SOURCES.md`.
3. **Self-Seeding (AD-12):** Jede Modul-Phase seedet ihre eigenen SHK-Defaults deklarativ (Pipelines, Custom-Felder, Templates, Rollen). Der Onboarding-Wizard (P20) orchestriert nur. Noah validiert die Seeds iterativ — blockiert den Bau nicht.
4. **EU-Stack ohne US-Kern-Abhängigkeit:** Postgres-RLS (AD-1), Hetzner (Hosting + S3-Object-Storage, Infra-ADR-009 — Scaleway abgelöst 12.06.), Meilisearch zentral (node1) + pgvector v2 (AD-5), Mistral als EU-LLM (AD-16), Per-Tenant-Verschlüsselung mit Crypto-Shredding (AD-8 revidiert: app-seitige Envelope-Encryption, Master-Key im Coolify-Secret-Store).

Wie eine Phasen-Seite zu lesen ist

Jede Phase (Abschnitt 4) hat fünf Blöcke: **Zweck** (was sie liefert) · **Umsetzung** (wie gebaut wird) · **Erwartung** (Definition of Done — beobachtbare Success Criteria) · **Abhängigkeiten** (braucht / liefert / integriert rückwirkend) · **Team-Hinweise** (Designer-Bedarf, offene Entscheidungen, Risiken).

---

2. Phasen-Übersicht

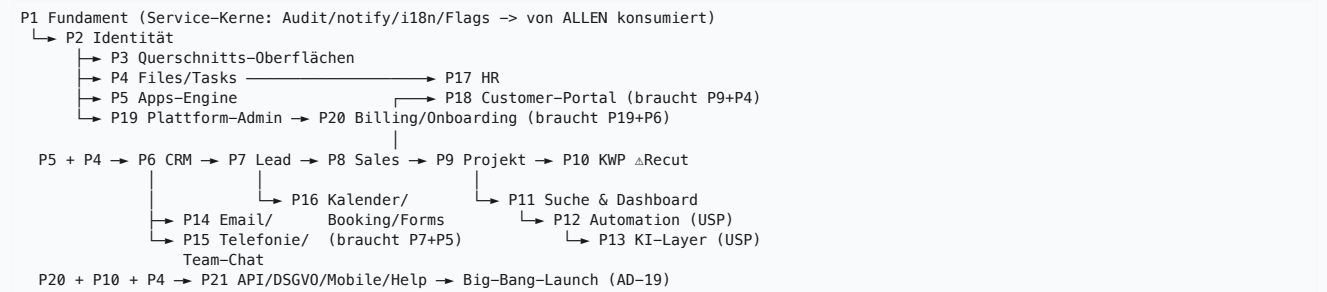
#	Phase	Liefert	Reqs	MS	Braucht	UI/Designer
1	Fundament & Walking-Skeleton	Codebasis, Hot-Swap-Maschine, Service-Kerne	11	15	—	nur Beispiel-Route
2	Identität & Mandantentrennung	Signup, Login, RLS, RBAC	4	9	1	ja (Auth-Screens)
3	Querschnitts-Oberflächen	Audit-Viewer, Notification-Center, Sprach-Settings (DE)	4	5	2	leicht
4	Files & Tasks	Datei-Verwaltung, Aufgaben	4	7	2	ja
5	Apps-Engine	Integrations-Framework (DataConnection)	2	3	2	mittel (Settings)
6	CRM-Kern	Kontakte, Firmen, Leads, Custom-Felder, Tags	9	5	5, 4	**schwer**
7	Lead-Inbox	Multi-Quellen-Triage, Anti-Spam	6	4	6	**schwer**
8	Sales-Pipeline	Deals, Kanban, Forecast	5	7	7	**schwer** (Kanban)
9	Auftrags-/Projektpipeline	Projekte, Plantafel, Offline-Abnahme	3	6	8	**schwer** (Plantafel)
10	~~KWP-Integration~~ → **v2 (erstes v2-Modul)**	API-Zugang ~19.07. > Launch	0	0	—	—
11	Suche & Dashboard	Keyword-Suche, Cmd+K, Standard-Widgets	5	5	9	**schwer**
12	Automation-Engine (USP)	Visueller Flow-Editor, Guards	5	8	11	**sehr schwer**
13	KI-Layer (USP)	EU-LLM-Gateway, Sprach-zu-Automation	1	5	12	mittel (Chat)
14	Email & Marketing	Versand+BCC-Brücke, Marketing voll	4	6	6, 5	**schwer**
15	Telefonie & Team-Chat	AirCall, Realtime-Chat	5	9	6, 5	**schwer**
16	Kalender, Booking, Forms, Signatur	Sync, Public-Booking, Form-Builder	10	15	7, 5	**sehr schwer**
17	HR-Hub	Verschlüsselte Personalprozesse	3	7	4	ja
18	~~Customer-Portal~~ → **v2**	deferred 11.06.	0	0	—	—
19	Plattform-Admin	Admin-Welt + Abo-Zusammenstellung	2	7	2	mittel (intern)
20	Billing, Onboarding & Branding	Stripe-Abo-Baukasten, Auto-Seed, Theme	4	8	19, 6	mittel
21	DSGVO & Launch-Reife	Export, Crypto-Shredding, Feedback	3	2	20, 10, 4	leicht

Summen v1: 90 Requirements · 133 Milestones (+ 43 committed in v2 — **KWP als erstes v2-Modul**, Mobile/Expo, Portal, Postfach-Sync, Public-API inkl. API-Keys, Help, HiveMind/Whisper, HalloPetra, Wizard-UI, AdminUser-Identität, Notification-Bündelung, Sprach-Wechsler u.a.; Historie: `.planning/SCOPE-V1-CUTS.md`).

---

3. Dependency-Landkarte

Ausführung: strikt numerisch 1 → 21 (Solo-Dev-Sequenz). Die Spalte „Braucht" zeigt die *echten* Daten-/Fähigkeits-Abhängigkeiten — daraus ergibt sich Umsortier-Spielraum: P14/P15 brauchen nur P6+P5, P17 nur P4, P19 nur P2.



Dependency-Inversion — die zwei Register (verbindlich, in `PHASE-SOURCES.md`):

- **Extension-Points (frühe Phase liefert):** P1 `notify()`-Kanäle inkl. ruhendem Push · P2 Subscription-Status-Stub · P4 polymorphe EntityLink + Task-Listen-Pattern · P7 `lead.qualified`-Event · P8 `deal.won`-Event + Aggregat-APIs · P9 Plantafel-VM-Slots · P13 lokales AiCredits-Konto · P18 Default-Theme · P19 generische Inbox-Quelle.
- **Retro-Integrationen (späte Phase besitzt):** P6 Task↔CRM-Links · P8 Auto-Verkaufsfall + Promotion-Modal-Option · P9 Won-Handoff · P10 KWP-Verdrahtung in P8/P9 · P11 Such-Upgrades + Sales-Widgets · P13 KI-Actions · P16 Kalender↔Plantafel + Abnahme-Templates · P17 Abwesenheiten→Plantafel · P20 Stripe ersetzt Stubs + Theme→Portal · P21 Push-Aktivierung + Portal-DSGVO + Helpdesk→Inbox.

Audit-Kontrakt: Vorwärts-Refs aus den 176 Milestone-Vorbedingungen MINUS Inversions-Register = **leer** (verifiziert 10.06.2026: 40/40 deckungsgleich).

---

4. Die 21 Phasen im Detail

Phase 1 — Fundament & Walking-Skeleton

Zweck: Das Repo ist eine lauffähige, EU-sterile Codebasis mit den Querschnitts-Neubauten, den vier Service-Kernen und den verbindlichen Integration-Contracts als Bau-Vertrag für alle 20 Folgephasen. Die wichtigste Phase des Projekts — hier entsteht die Maschine, auf der alles andere läuft.

Umsetzung:

- ToolCTRL-Boilerplate als Git-Downstream (upstream bleibt mergebar, AD-10); **EU-Sterilisation** (GitHub/Google-OAuth-Reste + `User.githubId/googleId` raus).
- **4 Querschnitts-Neubauten:** `logActivity()` + zentrale Activity-Tabelle, Search-Registry (`indexed-entities`), Subframe-Adapter-Layer, Mass-Action-Schutz.
- **4 Service-Kerne** (aus P3 vorgezogen — der gesamte Auth-Block braucht sie): AuditEvent mit SHA-256-Hash-Kette via Prisma-Middleware (default-on) · `notify()` mit Kanal-Abstraktion (In-App/Email, Push-Kanal ruhend bis P21) · `packages/i18n` mit Build-Lint gegen Hard-Strings + Resolution User→Tenant→DE · Feature-Flag-Engine mit `tenantHasFlag()` + Redis-Cache (Admin-UI folgt P19).
- **Hot-Swap-Maschine:** View-Model-Konvention (ESLint blockt React im VM-Paket), Action-Callback-Contract, `ui-mock` , Visual-Regression-Gate, Hot-Swap-Drill in CI.
- Beispiel-Route als Walking Skeleton; URL-Schemata (App/Admin/Portal/Forms/Booking/Help/API) dokumentiert.

Erwartung (Definition of Done): Boilerplate baut durch, upstream mergebar · EU-Sterilisation nachweisbar · Querschnittsdienste + Service-Kerne aufrufbar mit Smoke-Test · Beispiel-Route mit VM + Sidebar + grünem Smoke-Test gegen `ui-mock` · URL-Schemata als geprüfte Konvention.

Abhängigkeiten: Braucht: nichts. **Liefert:** alle Service-Kerne + Extension-Points, die jede Folgephase konsumiert. Quellen: `ARCHITECTURE-DECISIONS.md` , `INTEGRATION-CONTRACTS.md` , Service-Kern-Teile aus 4 Specs · **15 Milestones**.

Team-Hinweise: ⚠ Der ADR-Text sagt „Repo ist Docs-only" — real ist die Boilerplate seit 04.06. eingezogen. **Erster Planungsschritt: Ist-Stand erheben**, Erledigtes nicht neu bauen. ✅ AD-6 gelockt (Mailjet): notify()-Email-Kanal direkt anbindbar; Domain-Warm-up ab Tag 1. 🔌 NEU: Konnektor-Muster (Entitlement-Gate) entsteht hier. Designer: parallel startet Sprint D0 (Design-System + Archetypen) — kein Blocker füreinander.

Phase 2 — Identität & Mandantentrennung

> ⚠ ***Scope-Schnitt 12.06. (Boilerplate-Recheck):*** *API-Keys (M027) → v2 — wandern mit der Public-API; Boilerplate-* `ApiKey` *-Modell bleibt ungenutzt im Code. Auth-Block hat HIGH-Boilerplate-Coverage (Login/Magic-Link/TOTP fertig); Eigenbau-Kern ist* ***RLS*** *(Boilerplate hat nur App-Level-Scoping). MS 10→9, Reqs 5→4.*

Zweck: Ein Handwerksbetrieb registriert sich selbst, meldet sich sicher an und arbeitet in einem nachweisbar RLS-isolierten Mandanten mit rollenbasierten Rechten.

Umsetzung:

- Self-Service-Signup mit auto-generiertem Tenant-Slug (Reserved-Words geblockt) → `app.hivegrid.de`.
- Login Email+Passwort (NIST-Regeln, Breach-Check) + TOTP-2FA; Sessions serverseitig in Postgres (Geräte-Liste, Revoke, „überall abmelden", 4h-Inaktivität, 30-Tage-Sliding) + Redis-Cache (AD-4).
- **Postgres-RLS als bauliche Vorgabe** (AD-1): USING/WITH-CHECK-Policies, `SET LOCAL app.tenant_id`, App-Rolle ohne BYPASSRLS.
- RBAC mit 10 geseedeten SHK-Standard-Rollen; User-Lifecycle (Einladung 48h-Token, Deaktivierung, DSGVO-Löschung per PII-Anonymisierung). *(API-Keys → v2 mit Public-API.)*
- Extension-Point: **TenantSubscription-Status-Stub** (manuell setzbar) — der Frozen-Modus liest nur den Status; Stripe füllt ihn ab P20.

Erwartung: Signup→Tenant funktioniert · Login+2FA+Session-Verwaltung · Cross-Tenant-Zugriff nachweisbar unmöglich (RLS-Test) · Rollen geseedet, Lifecycle komplett.

Abhängigkeiten: Braucht: P1 (Service-Kerne: Login-Audit, Einladungs-Mails via notify(), Tarif-Gates via Flags). Liefert: Subscription-Stub. Quellen: Spec M27 · **9 Milestones**.

Team-Hinweise: ✅ AD-7 gelockt (mCaptcha self-hosted, infra/ vorhanden). AD-6 gelockt (Mailjet) für Einladungs-Mails. Designer: Auth-Screens (Login, Signup, 2FA, Geräte-Liste) — Standard-Formulare aus dem Form-Kit.

Phase 3 — Querschnitts-Oberflächen

> ⚠ ***Scope-Schnitt 12.06. (Boilerplate-Recheck):*** *Sprach-Wechsler (M029) + Notification-Bündelung/Quiet-Hours (M117) → v2; M028 gedünnt (**v1 = DE-only**, i18n-Wiring + CI-Gate bleiben Pflicht — alles i18n-ready); M152 gedünnt (fixe Retention, Legal-Hold/Saved-Views → v2). MS 7→5.*

Zweck: Die Bedien-Oberflächen der in P1 gebauten Plattformdienste: Audit-Viewer mit Export, Notification-Center mit Einstellungen, Sprach-Settings (DE-only).

Umsetzung:

- Audit-Viewer (gefiltert, permission-gegated) + NDJSON/CSV-Export ins Object Storage (Hetzner-S3) mit fixer Default-Retention (Boilerplate: Retention-Cron + PDF-Export vorhanden).
- Versand-Pipeline über BullMQ (In-App via SSE, Email; Push-Stub bis v2) — Sofort-Zustellung; Notification-Center mit Kategorie-An/Aus pro Kanal — **kanal-erweiterbar gebaut** (Push-Sektion folgt v2; Bündelung/Quiet-Hours = v2).
- Sprach-Settings DE-only; alle Strings durch `t()` mit vollständiger `de` -Locale (`verify:i18n` -Gate); `Contact.language` als Feld vorhanden (Outbound-Steuerung ab v2).

Erwartung: Audit-Viewer + Export funktionieren · Notification-Center mit Kategorie-Einstellungen, Pipeline stellt zu · DE-Locale vollständig, CI-Gate grün.

Abhängigkeiten: Braucht: P2 (User/Permissions für Settings). Die Service-Kerne selbst liegen in P1. Quellen: Specs M21/M22/M30 (Oberflächen-Teile) · **5 Milestones**. Retro-Empfänger: P11 rüstet die zentrale Suche im Audit-Viewer nach; v2 erweitert Settings um Push + Quiet-Hours.

Team-Hinweise: Leichte UI (List-Views + Settings aus Archetypen). Flag-Admin-UI liegt bewusst **nicht** hier, sondern in P19 (braucht Admin-Welt).

Phase 4 — Files & Tasks

> ⚠ ***Scope-Schnitt 12.06. (Boilerplate-Recheck):*** *M054/M055 gedünnt — Versionierung, Office-/Video-Previews, Resume-Upload, OCR → v2.* ***Bleibt Pflicht: HEIC→JPG-Konvertierung*** *(iPhone-Fotos der Monteure!). Permissions/Sharing, Tags, Archiv-Flag bleiben. MS-Zahl unverändert (7), Umfang schlanker.*

Zweck: Die nutzerseitigen Plattform-Primitive: Dateien an jede Entität hängen, Aufgaben mit Fälligkeit/Reminder verwalten.

Umsetzung:

- Chunked Upload (TUS) mit RLS auf Storage-Pfaden (Hetzner-S3, `tenants/{id}/`); Previews (PDF→PNG, Bild→WebP, **HEIC→JPG**); `FileEntityLink`.
- Tasks mit Status/Priorität/Fälligkeit, Watcher, Reminder via `notify()`, Auto-Archiv-Cron (30 Tage).
- Extension-Points: **polymorphe EntityLink-Tabelle** (`entity_type` + `entity_id`, kein harter FK — löst den Zyklus zu CRM: P6 verdrahtet die CRM-Entities später) und **Task-Listen-Pattern** als wiederverwendbare Komponente (Module P12/P14/P16 binden ihre Task-Listen selbst ein).

Erwartung: Upload+Preview (inkl. HEIC→JPG) · Datei↔Entität-Verknüpfung · Tasks mit Reminder · Auto-Archivierung.

Abhängigkeiten: Braucht: P2. Quellen: Specs M23/M26 · **7 Milestones** (M074 Task-ERP-Sync → P10, gehört zur KWP-Integration).

Team-Hinweise: Designer: Datei-Browser + Task-Listen (List-View-Archetyp). Customer-Portal-Upload wird hier vorbereitet (P18 nutzt ihn).

Phase 5 — Apps-Engine & Integrations-Framework

Zweck: EIN typsicheres Integrations-Framework, über das später alle externen Systeme laufen: Gmail, Outlook, AirCall, PandaDoc, Stripe, KWP.

Umsetzung:

- `DataConnection` -Datenmodell + Lifecycle; OAuth2-Connect-Flows und API-Key-Connect in der Settings-UI.
- Type-safe Adapter-Registry: neue Connection-Typen ohne Kern-Änderung registrierbar; `AppSyncRun` / `AppSyncItem` protokollieren jeden Sync-Lauf.
- Connection-Health-Monitoring; OAuth-Tokens per-Tenant-DEK verschlüsselt (Envelope, AD-8).

Erwartung: Connect-Flow (OAuth + API-Key) funktioniert end-to-end · Registry erweiterbar ohne Kern-Änderung · Health-Status sichtbar, Tokens verschlüsselt.

Abhängigkeiten: Braucht: P2. Quellen: Spec M33a · **3 Milestones**. Kleinste Phase — aber Vertrag für P10/P14/P15/P16/P20.

Team-Hinweise: Designer: Settings-/Connection-Cards-Archetyp. Wichtigste Qualitätsfrage: Typsicherheit der Registry (hier zahlt sich Sorgfalt 6× aus).

Phase 6 — CRM-Kern

Zweck: Vertriebler verwalten Kontakte, Firmen und Leads als typisierte First-Class-Entitäten — mit Custom-Feldern, Tags, Activity-Stream und Duplikat-Schutz. Ablöse des HubSpot-CRM.

Umsetzung:

- Dedizierte Prisma-Tabellen (Contact, Company, Lead, Relation, ContactEmail/Phone, CustomFieldDefinition/Value, Tag) mit RLS; der **Boilerplate-Entity-Builder wird entfernt, nicht angepasst** (AD-2).
- Listen server-seitig: Filter/Sortierung/Cursor-Pagination, Bulk-Actions, CSV-Export, Spalten-Config pro User.
- Detail-Views mit Inline-Edit, Activity-Stream (append-only Event-Sourcing, 18+ Typen), verknüpften Entitäten, Custom-Feldern; SHK-Custom-Felder als Seed.
- Live-Duplikat-Check (Hard-Match Email/Telefon, Soft-Match Name+Firma/PLZ) + Bulk-Merge-Tool; Lead-Promotion-Modal (4 Optionen, transaktional); `crm.*`-Permissions mit server-seitigem Owner-Filter (404 statt 403).
- **Retro-Integration:** Task↔CRM-Entity-Links verdrahten (EntityLink aus P4).

Erwartung: Alle Entitäten typisiert + RLS, Entity-Builder raus · Listen vollständig server-seitig · Detail mit Stream + Custom-Feldern · Duplikat-Schutz + Promotion transaktional · Owner-Permissions greifen.

Abhängigkeiten: Braucht: P5 (Activity/Verknüpfungs-Muster), P4 (Files/Tasks in Detail-Views). Quellen: Spec M01 · **5 Milestones**.

Team-Hinweise: Designer: **schwerster Standard-Cluster** — List-View-, Detail-View-, Modal-Archetyp entstehen hier und prägen die halbe App. Noah-Validierung der SHK-Custom-Felder.

Phase 7 — Lead-Inbox

Zweck: Leads aus allen Quellen landen in einem Triage-Arbeitsplatz, werden automatisch angelegt, gegen Spam geschützt und sind auswertbar.

Umsetzung:

- Inbox mit Status-Filter, Quick-Actions, Bulk-Operationen, Realtime-Refresh.
- Auto-Anlage aus Form-Submission, eingehender Email, AirCall-Inbound (Quellen-Adapter; volle Quellen folgen mit P14/P15/P16).
- Status-Workflow `new|qualified|converted|lost|spam` mit definierten Übergängen; DSGVO-Auto-Lösch-Cron; Rate-Limiting + Anomalie-Detection mit Quarantäne + Admin-Notification; Statistiken-Page (KPIs, Quellen, Trends).
- Extension-Point: **`lead.qualified`-Event** — der Auto-Verkaufsfall (M049) wird in P8 daran angeschlossen.

Erwartung: Triage-Arbeitsplatz komplett · Auto-Anlage + Automation-Trigger-Hooks · Workflow + DSGVO-Cron · Quarantäne · Statistiken permission-aware.

Abhängigkeiten: Braucht: P6 (Lead-Entity, Promotion). Quellen: Spec M02 · **4 Milestones** (M049 → P8).

Team-Hinweise: Designer: Inbox + Statistik-Dashboard. SHK-Kernpfad beginnt hier sichtbar zu werden — guter Zeitpunkt für erste Noah-Demos.

Phase 8 — Sales-Pipeline

Zweck: Deals laufen durch mehrere SHK-Pipelines mit Kanban, Forecast und feingranularer Sichtbarkeit von Volumen-Daten.

Umsetzung:

- Multi-Pipelines (Pipeline, PipelineStage, Deal, LostReason) mit fortlaufenden Deal-Nummern (HiveNumberSeq); **Self-Seeding der SHK-Pipelines + Verlust/Gewinn-Gründe**.
- Kanban mit Drag-&-Drop + Stage-Wechsel-Validation, Realtime via SSE (AD-9); Listen-View mit Filter/Bulk.
- **Auto-Verkaufsfall** (M049, aus P7 hierher): konsumiert das `lead.qualified`-Event; zusätzlich Deal-Anlage-Option ins Lead-Promotion-Modal (P6) verdrahten.
- Won-Handoff als **`deal.won`-Domain-Event** (P9 konsumiert); Auto-Archive-Cron; Forecast/KPIs/Stage-Conversion/Lost-Reasons mit Aggregat-Cache + **Aggregat-APIs** (P11 baut Widgets darauf); Field-Level-Permissions (`deals.read_volume_*`).

Erwartung: Pipelines geseedet + Deal-Nummern · Kanban + Liste mit Realtime · Auto-Verkaufsfall aus qualifiziertem Lead · Forecast/KPIs · Volumen-Ausblendung je Permission.

Abhängigkeiten: Braucht: P7. Liefert: `deal.won`, Aggregat-APIs. Quellen: Spec M03 · **7 Milestones**.

Team-Hinweise: Designer: Kanban-Board (eigener Bespoke-Aufwand). KWP-Sync-Hooks als no-op vorbereiten (P10 verdrahtet).

Phase 9 — Auftrags-/Projektpipeline

Zweck: Ein gewonnener Auftrag wird zum geplanten Projekt — mit Plantafel-Scheduling pro Monteur und digitaler, offline-fähiger Abnahme auf der Baustelle.

Umsetzung:

- ProjectPipeline/Project/ProjectMember/AcceptanceProtocol/ProjectTemplate; Projekte entstehen aus dem `deal.won`-Handoff (Retro-Integration dieser Phase).
- **Plantafel:** Disponent plant Aufträge pro Monteur/Tag; offline-fähig; **VM-Slots** für Termine/Abwesenheiten/Storno vorsehen (P16/P17 füllen sie).
- **Digitale Abnahme:** Protokoll mit Unterschrift, offline mit späterem Sync; v1 mit festem Formular (Form-Templates rüstet P16 nach). Abnahme-Defaults self-seeded.
- Material-Listen/Lager; KWP-Hooks (Archiv, Drift) als no-op — P10 verdrahtet.

Erwartung: Won-Handoff erzeugt Projekte · Plantafel offline-fähig · Abnahme offline mit Unterschrift + Sync.

Abhängigkeiten: Braucht: P8. Liefert: Plantafel-Slots, Abnahme v1. Quellen: Spec M04 · **6 Milestones**.

Team-Hinweise: Designer: **Plantafel ist Bespoke** (Custom-Grid, Touch, offline) — früh beauftragen. Offline-Verhalten (AD-17) hier zum ersten Mal ernst: IndexedDB-Queue-Muster entsteht hier und wird in P21 (Mobile) wiederverwendet.

Phase 10 — KWP-Integration ⚠ Recut — → v2 (ERSTES v2-Modul)

> 🔴 ***Scope-Schnitt #12 (12.06., Kalender-Realität):*** *KWP-API-Zugang lt. Hersteller-Mail erst ~**19.07.** — v1-Launch ist ~10.07. Das W3-Gate „KWP E2E" wäre nie haltbar gewesen. **KWP bleibt USP und wird das erste v2-Modul** — Geerkens hat den Sync ~2–3 Wochen nach Launch. Sofort erledigt: Callback-URL gemeldet (PKCE-Flow), bnWin.net-8.1-N3-Update angestoßen. v1 behält Apps-Engine + Extension-Points (Auftrag/Abnahme/Material). Beschreibung unten = v2-Planungsstand.*

Zweck: HiveGrid synchronisiert Auftrags- und Stammdaten bidirektional mit dem im SHK-Markt verbreiteten ERP **KWP** (Kernsystem des Erstkunden) — über die offizielle KWP-API. Damit ist die Kette **Lead → Sales → Auftrag → KWP durchgängig** (= Core Value erfüllt).

Umsetzung:

- KWP als `DataConnection type='rest-api'` in der Apps-Engine; **keine KWP-DB-Credentials in der Cloud** (KwpSettings, KwpRegistry, KwpProjectQueue, KwpApiKey).
- Bidirektionaler Sync mit Konfliktregel **„KWP gewinnt"** (AD-14/AD-17); Heartbeat/Recovery, Warteschlange/Replay.
- **Schema-Drift-Sentinel:** erkennt MSSQL-Schema-Änderungen per Snapshot-Vergleich, benachrichtigt Admin.
- **Retro-Integrationen dieser Phase:** Auftrag-Archiv-Sync (M066), Abnahme-Drift-Schutz (M069), Material-Listen (M070), Plantafel-Sync, Dokumenten-Archiv, Task-ERP-Sync (M074).

Erwartung: KWP-Connection ohne Cloud-Credentials · Sync bidirektional mit KWP-gewinnt · Drift-Sentinel meldet.

Abhängigkeiten: Braucht: P9 (Aufträge), P5 (Connection-Framework). Quellen: Spec M19 ⚠ · **8 Milestones**.

Team-Hinweise: 🔴 **Recut bindend:** Die M19-Spec beschreibt noch die obsolete Windows-Client-Architektur (`kwpsync.exe`). Die offizielle KWP-API macht M089 obsolet, M090–M095 vereinfachen sich, M093 bleibt. **M089–M095 werden in der Plan-Phase neu geschnitten, bevor Code entsteht.** Kein UI, kein Designer. Externer Faktor: KWP-API-Zugang/Doku von KWP beschaffen.

Phase 11 — Globale Suche & Dashboard

> ⚠ ***Scope-Schnitt 11.06.:*** *Custom-Widgets (M149) + Export (M150) → v2; Suche v1 = Keyword-only, pgvector/Hybrid → v2. MS 7→5.* ⚠ ***Ergänzung 12.06.:*** *M147 gedünnt — **3 feste Rollen-Layouts** (GF/Büro/Monteur) statt Drag&Drop-Grid; Widget-Registry bleibt voll (Sales-Widgets docken via Inversion an).*

Zweck: Jede Entität ist per Hotkey systemweit auffindbar; Kennzahlen stehen in einem rollenbasierten Widget-Dashboard.

Umsetzung:

- **Suche (AD-5 sequenziert):** zentrale Meilisearch auf node1 mit scoped Key (Keyword, v1) + pgvector-Embeddings via RRF-Fusion (→ v2 mit HiveMind); permission-aware über SearchAdapter; Index-Updates per Outbox-Pattern (Registry aus P1 ist seit Tag 1 befüllt).
- Cmd+K-Modal (debounced, RecentSearch) + Vollbild-Such-View.
- Widget-Dashboard rollenbasiert (SHK-Default-Layout self-seeded), Live-Updates (WS/SSE), Aggregat-Cache.
- **Retro-Integrationen:** Hybrid-Suche-Upgrade für Audit-Viewer (P3), Datei-Suche (P4), CRM-Segmente (P6), Kanban-Filter (P8); Sales-Reporting-Widgets auf P8-Aggregat-APIs.

Erwartung: Hybrid-Treffer permission-aware · Outbox-Indexierung · Cmd+K · Dashboard live + geseedet.

Abhängigkeiten: Braucht: P9 (Entitäten der Kette vorhanden). Quellen: Specs M24/M20 · **7 Milestones**.

Team-Hinweise: ⚠ **AD-5-Infra (self-hosted Meilisearch) vor Modulbau bereitstellen.** Designer: Command-Palette + Widget-Grid (Bespoke).

Phase 12 — Automation-Engine (USP-Säule 1)

Zweck: Der visuelle, KI-gestützte Automation-Editor über alle Module — in voller Tiefe (AD-16), nicht als Fundament-Light.

Umsetzung:

- Graph-/Flow-Editor; vollständige **Trigger-Library** (CRM, Sales, Projekte, Zeit, Webhooks; Email/Forms/Kalender-Trigger registrieren sich mit P14/P16) und **Action-Library** (CRM, Notifications, KWP-Push, HTTP, Tasks, If/Else, Delay, Loop) — **Registry type-safe erweiterbar** (KI-Actions folgen P13).
- Sprach-zu-Automation (DE), Dry-Run/Test-Lauf, **24h-Undo-Fenster**.
- Schutz: Loop-Guard, Rate-Limit, Idempotenz-Key, Resume-Mode, Trigger-Volumen-Metering pro Tenant; Klartext-Logs.
- Persistenz: AutomationRule (JSON-Graph), AutomationExecution, StepExecution, AutomationTemplate (self-seeded SHK-Templates).

Erwartung: Flows aus voller Trigger/Action-Library baubar · Sprache→Automation + Dry-Run + 24h-Undo · Runaway-Schutz komplett · Ausführungen nachvollziehbar persistiert.

Abhängigkeiten: Braucht: P11 (und transitiv die Domänen P6–P10). Quellen: Spec M17 · **8 Milestones**.

Team-Hinweise: Designer: 🔴 **Node-Editor = härteste UI der App** — längste Design-Vorlaufzeit, im Designer-Track früh starten (siehe Designer-First-Playbook D2). Sprach-zu-Automation nutzt das LLM-Gateway-Minimum; volle KI-Tiefe kommt P13.

Phase 13 — KI-Layer (USP-Säule 2)

> ⚠ ***Scope-Schnitt 11.06.:*** *HiveMind/RAG (M087), Whisper, AlephAlpha → v2 — v1 baut Andock-Adapter (RetrievalProvider, Outbox-Events, Bestätigungs-Grenze). Sprach-zu-Automation EXKLUSIV in der Automation. Reqs 4→1, MS 6→5.*

Zweck: Der EU-only KI-Layer in voller Tiefe: Multi-Provider-LLM-Gateway, RAG-Retrieval, Anruf-Transkription, HiveMind-Assistent.

Umsetzung:

- **EU-LLM-Gateway** (Mistral La Plateforme primär; Multi-Provider-Abstraktion, zweiter EU-Provider → v2 — Infra-Wechsel 12.06.: Scaleway entfällt); Token-Tracking; **lokales AiCredits-Konto** mit Deduktion (Kauf/Abrechnung folgt P20).
- RAG-Pipeline: pgvector, Outbox-Embedding-Jobs, **RLS-gefiltertes** Top-K-Retrieval.
- HiveMind-Assistent: kontextuelle RAG-Chats mit Conversation-History; **Promptfoo-Eval-Suite** als Sicherheits-/Qualitäts-Gate.
- Whisper-Transkription über EU-Endpoint (Anruf-Aufzeichnungen aus P15, retro angeschlossen).
- **Retro-Integration:** KI-Actions in der Automation-Registry registrieren (Sprach-zu-Automation wird voll).

Erwartung: Gateway EU-only mit Fallback + Credits-Deduktion · RAG RLS-gefiltert · HiveMind mit History + Promptfoo-Tests · Whisper EU.

Abhängigkeiten: Braucht: P12. Quellen: Spec M18 · **6 Milestones**.

Team-Hinweise: Designer: Chat-UI (mittel). Promptfoo-Suite ist Pflicht-Gate (A.14) — nicht streichen. Token-Kosten-Modell mit Noah/Pricing-Runde abstimmen (Preise geparkt, AD-15).

Phase 14 — Email & Marketing

> ⚠ ***Scope-Schnitt 11.06.:*** *Postfach-OAuth-Sync (M096–098) → v2. Bleiben: Sende-Domain-Verifikation, Mailjet-Versand,* ***BCC-Logging-Brücke (neu)****, Marketing VOLL inkl. A/B. MS 9→6, Reqs 6→4.*

Zweck: Der Betrieb arbeitet seinen Email-Verkehr in HiveGrid ab (bidirektionale Gmail/Outlook-Inbox mit CRM-Matching) und versendet DSGVO-konforme Marketing-Kampagnen.

Umsetzung:

- Gmail/Outlook-Sync bidirektional über Apps-Engine-Connections; Thread-View, Compose, Vorlagen; Auto-Matching Absender → Contact/Lead; Tracking/Bounces; Permission-Scoping.
- Marketing: Editor, Segmente/Empfänger, Versand-Engine über BullMQ mit Rate-Limiting + Bounce-Handling; **Double-Opt-In, Suppression-Check, One-Click-Unsubscribe** (DSGVO-Pflicht).
- SMTP via **Mailjet** (AD-6 gelockt 11.06.) — keine US-Kern-Abhängigkeit. Scope-Schnitt: Postfach-OAuth-Sync → v2, BCC-Brücke neu.
- Self-Seed Email-/Kampagnen-Templates; Retro: Task-Liste im Marketing-Modul; M068-Channel-Bedarf prüfen (voraussichtlich durch P1-notify() gedeckt).

Erwartung: Inbox bidirektional + CRM-Matching · Kampagnen DSGVO-konform · Versand über EU-Provider-Abstraktion.

Abhängigkeiten: Braucht: P6 (Contacts), P5 (Connections). Quellen: Specs M09/M10 · **9 Milestones**.

Team-Hinweise: ⚠ Spätestens hier muss **AD-6 final entschieden** sein. Designer: Inbox-Thread-View + Kampagnen-Editor (schwer).

Phase 15 — Telefonie & Team-Chat

> ⚠ ***Scope-Schnitt 11.06.:*** *HalloPetra (M105) → v2; AirCall + Anruf-Trigger + Chat komplett bleiben. MS 10→9.*

Zweck: Telefonie und interne Kommunikation laufen in HiveGrid: Click-to-Call/Inbound über AirCall mit Auto-Lead-Anlage und Realtime-Team-Chat mit entity-linked Channels. Ablöse von Quiply.

Umsetzung:

- AirCall-Adapter (Apps-Engine, verschlüsselte Tokens): Click-to-Call, Inbound-Routing, Voicemail/Notizen; unbekannte Nummer → Auto-Lead; HalloPetra-Anbindung.
- Team-Chat: **WebSocket-Room-Layer** (AD-9) mit Presence, Typing, Read-Receipts über Redis-Pub/Sub; 3 Channel-Typen (public, private, **entity-linked** im Deal-/Projekt-Detail).
- @Mentions mit Push-Hook, File-Uploads (via P4), Link-Previews, Chat-Verlaufs-Suche.

Erwartung: Calls über AirCall + Auto-Lead · Chat realtime mit allen drei Channel-Typen · Mentions/Uploads/Previews/Suche.

Abhängigkeiten: Braucht: P6, P5. Quellen: Specs M12/M13 · **10 Milestones**.

Team-Hinweise: Designer: Chat-UI + Call-Widgets (schwer). M11 WhatsApp ist v2 — Referenzen aus M13 als Stub markieren. Anruf-Aufzeichnungen → Whisper (P13) retro.

Phase 16 — Kalender, Booking, Forms & Signatur

> ⚠ ***Scope-Schnitt 11.06.:*** *Kalender v1 nur* ***Google*** *(Outlook → v2); Booking mit einer Zuteilungs-Strategie („Erster der frei ist"). Forms + Signatur bleiben voll.*

Zweck: Termin- und Dokument-Workflows: bidirektionaler Kalender-Sync, öffentliche Buchungsseiten, Drag-&-Drop-Form-Builder, Multi-Signer-Angebots-Signatur.

Umsetzung:

- CalendarAdapter Google/Outlook (Auto-Token-Refresh, Webhook-Subscriptions), Termin-Sync + Anzeige.
- **Public Booking:** Slot-Berechnung timezone-aware (Verfügbarkeitsregeln, Ausnahmen, Puffer), iFrame-Embed; Buchung → Lead + Activity + Bestätigungs-Mail; Stornierung.
- **Form-Builder:** 12 Feld-Typen, Conditional Logic, SHK-Templates (self-seeded), iFrame-Embed; Post-Submit-Actions (Lead/Deal/Task, Webhook, Email, Automation-Trigger) mit Field-Mapping + ActionLog; Rate-Limit/Captcha/DSGVO.
- Signatur: Multi-Signer via PandaDoc/DocuSign über Apps-Engine, Webhook-Verarbeitung, Status-Tracking.
- **Retro-Integrationen:** Kalender in Plantafel (P9) + Termin aus Won-Deal (P8); Form-Templates für die Abnahme (P9); Task-Liste im Forms-Modul.

Erwartung: Kalender-Sync bidirektional · Booking public end-to-end · Form-Builder mit Actions · Signatur-Workflow mit Status.

Abhängigkeiten: Braucht: P7 (Lead-Anlage aus Booking/Forms), P5. Quellen: Specs M25/M06/M05/M07 · **15 Milestones** (größte Phase).

Team-Hinweise: Designer: 🔴 **Form-Builder-Canvas = zweitschwerste UI**; Booking-Seiten sind **extern sichtbar** (Polish + Branding). ⚠ AD-7 (Captcha) hier zwingend entschieden.

Phase 17 — HR-Hub

> ⚠ ***Scope-Schnitt 11.06.:*** *Memberspot/Boobora-Onboarding (M138) → v2; Surveys bleiben. MS 8→7.*

Zweck: Interne Personalprozesse in einem verschlüsselten HR-Hub: Personalakten, Passwort-Safe, Urlaubs-/Krank-Workflow.

Umsetzung:

- HR-Profile, Urlaubs-/Krank-/Payslip-/Request-Daten **AES-256-GCM-verschlüsselt**; sensible Felder mit app-seitiger Envelope-Encryption (AD-8 revidiert 12.06.: Master-Key im Coolify-Secret-Store statt Cloud-KMS).
- Verschlüsselter Passwort-Safe pro Mitarbeiter mit Audit-Trail + DSGVO-Export.
- Mehrstufiger Urlaubs-Approval-Workflow; Krankmeldungs-Erfassung; Abwesenheits-Kalender.
- **Retro-Integration:** Abwesenheiten in der Plantafel (P9) anzeigen. Mobile-Selbstbedienung folgt P21.

Erwartung: HR-Daten verschlüsselt (nachweisbar) · Safe mit Audit · Approval-Workflow + Abwesenheits-Kalender.

Abhängigkeiten: Braucht: P4 (Dokumente). Quellen: Spec M14 · **8 Milestones**.

Team-Hinweise: Designer: HR-Screens (Standard-Archetypen). Verschlüsselung hier ist Muster für DSGVO-Shredding (P21).

Phase 18 — Customer-Portal

> 🔴 ***Scope-Schnitt 11.06.: Diese Phase ist KOMPLETT → v2 verschoben*** *(AD-18 revidiert). Die Beschreibung unten gilt für die v2-Planung. v1 behält die Andockpunkte (FileEntityLink-Vorbereitung, portal-URL-Schema).*

Zweck: Die externe Endkunden-Welt: ein vollständig von der Tenant-User-Welt getrenntes Portal mit Magic-Link-Login und explizit freigegebener Projekt-Sicht.

Umsetzung:

- **Getrennte Auth-Welt:** PortalUser/CustomerPortalSession (One-Time-Token-Magic-Link, IP-Bindung) — kein Durchgriff auf Tenant-User (Prinzip AD-3); PortalConfiguration.
- Projekt-Sicht über CustomerPortalProjectAccess: **Default NICHT sichtbar**, nur explizit freigegebene Projekte.
- Dokument-Upload (via P4-Vorbereitung), Portal-Forms, Reklamation; Subunternehmer-Zugang; Skalierungs-Sicht (Service-Inbox + Routing, AD-18); startet mit Default-Theme (P20 rüstet Branding nach).

Erwartung: Magic-Link-Login getrennt · nur freigegebene Projekte sichtbar · Upload + Portal-Forms; DSGVO-Lebensdauer-Logik vorbereitet (P21 verdrahtet Löschung).

Abhängigkeiten: Braucht: P9 (Projekte), P4 (Files). Quellen: Spec M15 · **6 Milestones**.

Team-Hinweise: Designer: **extern sichtbar → höchster Polish-Anspruch**, eigene Design-Sprache (Playbook D4). Sicherheits-Review der Auth-Trennung einplanen.

Phase 19 — Plattform-Admin

> ⚠ ***Scope-Schnitt 11.06.:*** *Flag-Massen-Statistik (M033) + Plattform-Lead-Pipeline (M040) → v2.* ***NEU:*** *Admin stellt Tenant-Abos zusammen (Feature-Konnektoren schalten + Stripe-Positionen — Abo-Baukasten-Mechanik). MS 11→9. ⚠* ***Ergänzung 12.06. (Boilerplate-Recheck):*** *Getrennte AdminUser-Identität/AdminRBAC (M038/M039) → v2 — v1 nutzt das Boilerplate-Modell (User-Extension + MFA-Step-up).* ***Bindende Auflagen:*** *TOTP-Pflicht · Admin-User* ***ohne Tenant-Workspace*** *· Host-Routing* `admin.hivegrid.de` *+* ***Cookie-Domain-Trennung*** *(Code routet bisher* `/admin` *-Pfad). M035 gedünnt (Boilerplate-KPIs; Churn/ARPU → v2). MS 9→7.*

Zweck: Die Plattform ist aus einer getrennten Admin-Welt betreibbar: Tenant-Management, Subscription-Overrides, Feature-Flag-Verwaltung, Queue-Monitoring, Support-Inbox.

Umsetzung:

- **Vollständig getrennte Admin-Identität (AD-3):** AdminUser/AdminSession (IP-gebunden)/AdminRole-RBAC auf `admin.hivegrid.de` ; Login Email+PW+TOTP (Magic-Link verboten), optional Authentik-SSO; **kein Generalschlüssel** zwischen Plattform und Mandanten; Stellvertreter-Anmeldung auditiert.
- Tenant-Verwaltung, Subscription-Overrides (liest P2-Stub; Stripe füllt ab P20), **Flag-Admin-UI** (M031–M033: Schalter/Lifecycle, Filter-Editor, Massen-Statistik — Engine liegt seit P1), Lead-Pipeline für Plattform-Vertrieb, System-Health + BullMQ-Monitoring, **Support-Inbox mit generischer Message-Quelle** (P21 schließt Help-Desk an).
- AdminAuditLog = vorhandener AuditEvent mit `tenantId=null` (Hash-Kette aus P1).

Erwartung: Getrennte Admin-Welt nachweisbar · Tenants/Overrides/Flags zentral verwaltbar · Queues + Support-Inbox laufen, alles auditiert.

Abhängigkeiten: Braucht: P2 (Tenants existieren). Quellen: Specs M37 + M32 (Admin-UI-Teile) · **11 Milestones**.

Team-Hinweise: Designer: internes Admin-UI (mittel, Archetypen reichen). Kann dank Dep nur auf P2 bei Bedarf vorgezogen werden.

Phase 20 — Billing, Onboarding & Branding

> ⚠ ***Scope-Schnitt 11.06.:*** *Wizard-UI (M157) → v2 (Signup → Auto-SHK-Seed).* ***M154-Recut:*** *Abo-Baukasten (Basis pro Nutzer + Feature-Positionen) statt Seat-Typen — Legacy-Referenz* `pricing-plans.ts` *(Core/Swarm/Colony). M156: Zähler+Drossel ja, Aufschlags-Abrechnung → v2. MS 9→8.*

Zweck: Ein neuer Betrieb kommt vom Signup über einen branchen-getriebenen Wizard ins Produkt, läuft auf Stripe-Billing, wählt sein Theme.

Umsetzung:

- **Stripe:** TenantSubscription real (ersetzt P2-Stub — Retro-Integration), Webhook-Verarbeitung, Plan/Seat-Modell, 14-Tage-Trial ohne Karte, Dunning Tag 1/3/7, **Frozen-Modus** (Read-Only-Middleware); **AI-Credits-Kauf + Metering** (Konto aus P13); konkrete Preis-Stufen bleiben **geparkt** (AD-15), Geerkens auf Sonderkondition.
- **Onboarding-Wizard** mit SHK-Vorauswahl (OnboardingSession): orchestriert die in den Modul-Phasen vorhandenen Seeds; optionale Demo-Daten (5 Contacts, 3 Deals, 2 Projekte) via `is_demo_data` rückstandsfrei löschbar.
- **Theme-Preference** Light/Dark/System flash-frei (SSR-Cookie); Retro: Theme aufs Customer-Portal anwenden. (Tenant-White-Label/Custom-Domain = v2.)

Erwartung: Trial→Dunning→Frozen end-to-end · Wizard provisioniert SHK-Tenant · Demo-Daten löschbar · Theme flash-frei.

Abhängigkeiten: Braucht: P19 (Admin-Overrides), P6 (Demo-Daten-Entities). Quellen: Specs M28/M29/M31 · **9 Milestones**.

Team-Hinweise: Designer: Onboarding-Wizard ist conversion-kritisch (Wizard-Archetyp). Stripe-Webhooks gegen Test-Clock testen.

Phase 21 — DSGVO & Launch-Reife

> 🔴 ***Scope-Schnitt 11.06.:*** *Public-API (M161–163), Mobile-App (M167–171 → v2 mit **Expo** statt Capacitor), Help-Center/Desk (M172–175), DSGVO-Auskunfts-Tooling (M166 → manuelles Playbook) — alles → v2.* ***Bleiben:*** *DSGVO-Export + Lösch-Workflow (Crypto-Shredding), Feedback-Widget → Admin-Inbox, Minimal-FAQ; Monteure via PWA. MS 15→2, Reqs 9→3. Beschreibung unten teilweise v2-Stand.*

Zweck: Launch-Reife: öffentliche Read-API + Webhooks, kompletter DSGVO-Lebenszyklus, Capacitor-Offline-App, Help-Center + Feedback. **Danach ist der Big-Bang-Launch (AD-19) auslösbar.**

Umsetzung:

- **Public-API:** Read-Only-REST (Contacts, Deals, Leads, Projects, Tasks, Companies) mit API-Key-Auth (M027 — seit 12.06. ebenfalls v2, wandert als Paket mit), Rate-Limit, OpenAPI-Spec; eingehende Lead-Webhooks + ausgehende HMAC-signierte Subscriptions mit Retry/Logging.
- **DSGVO:** ZIP-Export (72h-signed-URL); Tenant-Löschung mit **Crypto-Shredding** (Per-Tenant-DEK verwerfen, AD-8), Stripe-Cancel, Storage/Cache-Bereinigung; Auskunfts-Workflow; **Retro:** Portal-User-Löschung (P18).
- **Mobile (Capacitor, iOS ≥16 / Android ≥10):** Biometrie-Lock, Kamera-Upload, Deeplinks, **voller Offline-Betrieb** (IndexedDB, OfflineSyncQueue, „KWP gewinnt", AD-17), App-Store-Compliance; **Push via APNs/FCM** — aktiviert den ruhenden notify()-Kanal aus P1 und erweitert retro die Settings (P3), HR-Selbstbedienung (P17), Theme (P20).
- **Help:** Help-Center (dateibasiertes Markdown, FlexSearch, SSR-SEO, Deep-Links); In-App-Feedback-Widget (Screenshot, Rate-Limit) → speist die Admin-Support-Inbox (P19, Retro); Help-Desk-Ticket-Eingang + Self-Service.

Erwartung: API + Webhooks produktiv · DSGVO-Export + Crypto-Shredding nachweisbar · Mobile-App offline-fähig mit Push · Help live → **Launch-Checkliste komplett**.

Abhängigkeiten: Braucht: P20 (Billing für Lifecycle), P10 (KWP-Konfliktregel für Offline-Sync), P4 (Uploads). Quellen: Specs M34/M35/M36/M38/M39 · **15 Milestones**.

Team-Hinweise: Designer: komplette Mobile-UI (eigenes Paradigma, Playbook D4) + Help-Site. App-Store-Review-Zeiten in den Launch-Plan einrechnen.

---

5. Kritische Punkte fürs Team

Offene Entscheidungen (blockieren punktuell, jetzt terminieren):

1. ✅ ~~AD-6~~ **GELOCKT (11.06.): Mailjet** (Boilerplate-verdrahtet). Verbleibend: Sende-Domain-Setup + **Warm-up ab Tag 1** — Marketing-Kampagnen (voll in v1!) brauchen Domain-Reputation.
2. ✅ ~~AD-7~~ **GELOCKT (11.06.): mCaptcha self-hosted** (`infra/mcaptcha/` liegt bereit) — Login (P2), Public-Forms/Booking (P16).
3. 🟠 **AD-5 Meilisearch-Infra:** self-hosted Instanz vor P11 (Ops). pgvector/Hybrid → v2 (sequenziert 11.06.).

Bekannte Risiken / Pflichten:

4. 🔴 **KWP-Recut (P10):** M089–M095 vor Code-Entstehung neu schneiden — offizielle KWP-API statt `kwpsync.exe`. Externer Faktor: API-Zugang + Doku von KWP.
5. 🟠 **ADR ↔ Repo-Realität (P1):** ADR-Text sagt „Docs-only", real ist die Boilerplate seit 04.06. eingezogen → P1-Planung beginnt mit Ist-Stand-Erhebung.
6. 🟡 **Spec-Status:** Alle 36 Specs sind `detail-spec-draft` → vor jedem Modulbau: Noah-Verifizierung, Index-/Migrations-SQL, SHK-Defaults, Subframe-Designs.
7. 🟡 **M11 WhatsApp** ist v2 (nicht spec'd), wird von M13/M18 referenziert → in P13/P15 als Stub markieren.
8. 🟡 **Big-Bang-Risiko (AD-19):** spätes Nutzer-Feedback ist bewusst akzeptiert — die Gegenmaßnahme (Noah-Klick-Pfad-Validierung **pro Modul**) ist Pflicht, nicht optional. Erste Demos sinnvoll ab P7/P8.

Designer-Track (Kurzfassung, Details im Designer-First-Playbook): Sprint D0 (Design-System + Archetypen) parallel zu P1 → CRM-Cluster (P6) → die vier Bespoke-Langläufer **sofort** starten: Flow-Editor (P12), Form-Builder (P16), Plantafel (P9), Dashboard/Cmd+K (P11). Extern sichtbare Flächen (P16-Booking, P18-Portal, P21-Mobile/Help) bekommen eine eigene Politur-Runde.

6. Glossar

Begriff	Bedeutung
View-Model (VM)	Reines Daten-Objekt aus dem Route-Loader; UI-Komponenten erhalten nur VM + Callbacks. ESLint verbietet React im VM-Paket.
Subframe-Adapter	Dünne Mapping-Komponente, die einen Subframe-Design-Screen an VM-Slots bindet — Designer-Output bleibt unangetastet.
ui-mock / Hot-Swap-Drill	Styleless Ersatz-UI; CI tauscht das echte UI dagegen aus — läuft die App weiter, ist die Trennung bewiesen.
Extension-Point	Andockpunkt (Event, Stub-Tabelle, VM-Slot), den eine frühe Phase liefert, damit eine spätere sich integrieren kann.
Retro-Integration	Integrations-Aufgabe, die der späteren Phase gehört (Dependency-Inversion) — beim Planen MIT einzuplanen.
Self-Seeding	Jede Modul-Phase legt ihre SHK-Defaults (Pipelines, Felder, Templates) selbst deklarativ an; der Onboarding-Wizard orchestriert nur.
AD-x	Architektur-Entscheidung aus `docs/ARCHITECTURE-DECISIONS.md` (AD-6 Mailjet + AD-7 mCaptcha am 11.06. gelockt; AD-5 sequenziert, AD-17/18 revidiert).
Milestone (M0xx)	BMAD-Detail-Plan unter `docs/milestones/` — die Plan-Ebene unterhalb der Phasen (176 Stück, je Phase zugeordnet).
Walking Skeleton	Dünnster end-to-end lauffähiger Durchstich (P1): Route → VM → DB + Test, an dem alle Konventionen hängen.

*Aktualisiert: 12.06.2026 nach Scope-Schnitt v1/v2 — v1 = 146 MS / 94 Reqs, 30 MS committed in v2 (Historie: `.planning/SCOPE-V1-CUTS.md`). Neu: Konnektor-Pflicht je Modul + Abo-Baukasten-Billing (Legacy-Referenz Core/Swarm/Colony). Maschinenlesbare Quellen: `.planning/` im Repo. Aktualisiert: 12.06.2026 (zweiter Pass) nach Boilerplate-Recheck-Schnitt #11 — **v1 = 141 MS / 93 Reqs, 35 MS committed in v2**. → v2: API-Keys (M027), Sprach-Wechsler (M029, v1 DE-only), AdminUser-Identität (M038/M039, Auflagen an Admin-Auth), Notification-Bündelung (M117). Gedünnt: M028/M035/M054 (HEIC→JPG bleibt)/M055/M147 (feste Layouts)/M152. Audit: 141+35 = 176 ✓, Register 31/31, 0 unaufgelöste Vorwärts-Deps. Aktualisiert: 12.06.2026 (vierter Pass) — **Infra-Wechsel eingearbeitet** (Infra-Repo ADR-009/010): Hetzner + Coolify + Cloudflare statt Scaleway/Fly; Storage = Hetzner-S3, LLM = Mistral primär, AD-8 revidiert (app-seitige Envelope-Encryption); Meilisearch/mCaptcha/ClamAV/Sentry laufen bereits zentral. Arbeitsrhythmus: Mo–So ≥8h → 2,3 MS/Dev/Tag bei 2 Devs. Aktualisiert: 12.06.2026 (dritter Pass) nach Scope-Schnitt #12 — **v1 = 133 MS / 90 Reqs, 43 MS committed in v2. P10 KWP komplett → v2 als ERSTES v2-Modul** (Hersteller-API-Zugang ~19.07. > Launch ~10.07.; Callback-URL `app.hivegrid.de/api/apps/kwp/oauth/callback` gemeldet, bnWin.net 8.1 N3 angestoßen). Gedünnt: M030 (nur Hell/Dunkel), M070 (Lagerist → v2, Noah-Frage), M122 (personalisierte Links → v2), M160 (KI-Walkthrough → v2); M129 bleibt voll. Audit: 133+43 = 176 ✓, Register 28/28, 0 unaufgelöst. W4 ohne KWP — Dev A frei für E2E-Härtung.*