

Fulfillment Backend Development Plan (Supabase)

— Version 5

Owner: Jonathan Anderson (President Anderson)

Context: Aligns to *Fulfillment Display v5* and the Jonathan Anderson — Master Brief (Oct 7, 2025).

Stack: Supabase (Postgres + Auth + Storage + Edge Functions), TypeScript, Drizzle/Prisma (optional), pg audit, pg_cron, Stripe (optional), GitHub Actions CI/CD.

1) Objectives & Scope

- Provide a production-ready backend and “super-based databasing” (Supabase) for Fulfillment v5.
- Model core entities: Areas of Fulfillment, Dimensions, Metrics, Journals (incl. Autobiography), Integrity Recovery Log, Forgiveness & Reconciliation Journal, Rituals & Practices, Relationships Graph, Work & Finance Integration, Holistic Stability Dashboard, and Interpretation Keys.
- Ship secure APIs with granular RLS, migrations, seed, and observability so agents can begin coding immediately.

Non-Goals (for v1.0): UI/UX frontends; heavy analytics ML; third-party syncs beyond basic webhooks.

2) High-Level Architecture

- **Supabase Project:** Postgres 15+, GoTrue Auth (email + OAuth), Row Level Security enforced by default.
 - **Edge Functions:** TypeScript functions for heavy/secure ops (webhooks, batch compute, journaling NLP hooks).
 - **API Surface:**
 - **REST** via Supabase auto-generated endpoints (tables & views).
 - **RPC** via Postgres functions for complex actions (e.g., rollups, snapshotting).
 - **Storage:** Buckets for attachments (evidence, media, PDFs), per-resource prefixes, signed URLs.
 - **Jobs:** `pg_cron` for scheduled rollups (daily/weekly/monthly) and backups; optional Queue using `pgmq`.
-

3) Environments & Operational Standards

- **Envs:** `dev`, `staging`, `prod` — isolated Supabase projects; seed data only in `dev` / `staging`.
- **Secrets:** Managed via Supabase dashboard + GitHub Actions OIDC; no secrets in repo.
- **Migrations:** SQL first (db/supabase/migrations). Use `supabase db push` + codegen.
- **Branching:** `main` (prod), `develop` (staging), feature branches via PRs.
- **Observability:** pg audit enabled; logs shipped to Logflare/Datadog; structured JSON logs from edge functions.

4) Data Model (ER Overview)

```
users (supabase.auth) → profiles (1:1)
profiles → areas (1:N)
areas → dimensions (1:N)
dimensions → metrics (1:N)
metrics → metric_entries (1:N)
profiles → journals (1:N) → journal_entries (1:N)
journal_entries ↔ tags (M:N) via journal_entry_tags
profiles → rituals (1:N) → ritual_sessions (1:N)
profiles → forgiveness_logs (1:N) / reconciliation_logs (1:N)
profiles → integrity_events (1:N)
profiles → relationships (1:N) & relationship_events (1:N)
profiles → goals (OKR-style) → goal_updates
profiles → interpretations (auto-interpretation keys)
profiles → dashboards (1:N) → dashboard_snapshots
work_finance_integrations (per profile) → wfi_periods (monthly/quarterly)
attachments (polymorphic) linked to journal_entries, events, goals, etc.

Cross-cutting: audit_log, notifications, webhooks, api_keys.
```

5) SQL Schema (Core) — Run as migrations

Note: All tables ship with RLS = ON and policies below. UUID primary keys. Timestamps in UTC with `created_at`, `updated_at`. Soft-delete via `deleted_at` where noted.

```
-- 5.1 Profiles
create table public.profiles (
  id uuid primary key references auth.users(id) on delete cascade,
  display_name text,
  locale text default 'en-CA',
  timezone text default 'America/Toronto',
  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

-- 5.2 Areas of Fulfillment (aligned to v5)
create type area_kind as enum (
  'WORK_PURPOSE',
  'MUSIC',
  'WRITING',
  'PUBLIC_SPEAKING',
```

```

    'LEARNING_GROWTH',
    'FRIENDSHIP',
    'COMMUNITY',
    'FAMILY',
    'HEALTH',
    'SPIRITUAL',
    'FINANCE',
    'LAW_JUSTICE'
);

create table public.areas (
  id uuid primary key default gen_random_uuid(),
  profile_id uuid not null references public.profiles(id) on delete cascade,
  kind area_kind not null,
  title text not null,
  emoji text,
  color text,
  position int default 0,
  created_at timestamptz default now(),
  updated_at timestamptz default now(),
  unique(profile_id, kind)
);

-- 5.3 Dimensions (per area)
create table public.dimensions (
  id uuid primary key default gen_random_uuid(),
  area_id uuid not null references public.areas(id) on delete cascade,
  title text not null,
  description text,
  emoji text,
  color text,
  position int default 0,
  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

-- 5.4 Metrics & Entries
create table public.metrics (
  id uuid primary key default gen_random_uuid(),
  dimension_id uuid not null references public.dimensions(id) on delete cascade,
  key text not null,
  unit text default 'score',
  kind text check (kind in
('scalar', 'boolean', 'percent', 'enum', 'duration', 'money')) default 'scalar',
  min_value numeric,
  max_value numeric,
  enum_options text[],
  target numeric,

```

```

        created_at timestamptz default now(),
        updated_at timestamptz default now(),
        unique(dimension_id, key)
    );

create table public.metric_entries (
    id uuid primary key default gen_random_uuid(),
    metric_id uuid not null references public.metrics(id) on delete cascade,
    value_numeric numeric,
    value_text text,
    value_bool boolean,
    as_of date not null,
    note text,
    created_at timestamptz default now(),
    updated_at timestamptz default now(),
    unique(metric_id, as_of)
);

-- 5.5 Journals (incl. Autobiography linkage)
create type journal_kind as enum
('GENERAL', 'AUTOBIOGRAPHY', 'INTEGRITY', 'FORGIVENESS', 'RECONCILIATION');

create table public.journals (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid not null references public.profiles(id) on delete cascade,
    kind journal_kind not null default 'GENERAL',
    title text,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);

create table public.journal_entries (
    id uuid primary key default gen_random_uuid(),
    journal_id uuid not null references public.journals(id) on delete cascade,
    content markdown not null,
    mood int check (mood between 1 and 10),
    related_area uuid references public.areas(id),
    related_dimension uuid references public.dimensions(id),
    occurred_at timestamptz not null default now(),
    created_at timestamptz default now(),
    updated_at timestamptz default now(),
    deleted_at timestamptz
);

create table public.tags (
    id uuid primary key default gen_random_uuid(),
    name citext unique
);

```

```

create table public.journal_entry_tags (
  journal_entry_id uuid references public.journal_entries(id) on delete cascade,
  tag_id uuid references public.tags(id) on delete cascade,
  primary key (journal_entry_id, tag_id)
);

-- 5.6 Integrity / Forgiveness / Reconciliation Events
create table public.integrity_events (
  id uuid primary key default gen_random_uuid(),
  profile_id uuid not null references public.profiles(id) on delete cascade,
  title text not null,
  description text,
  category text check (category in ('breach', 'repair', 'commitment', 'boundary'))
not null,
  status text check (status in ('open', 'in_progress', 'resolved')) default
'open',
  occurred_at timestamptz not null,
  resolved_at timestamptz,
  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

create table public.forgiveness_logs (
  id uuid primary key default gen_random_uuid(),
  profile_id uuid not null references public.profiles(id) on delete cascade,
  subject text not null,
  narrative markdown,
  stage text check (stage in ('acknowledge', 'feel', 'release', 'recommit')),
  occurred_at timestamptz default now(),
  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

create table public.reconciliation_logs (
  id uuid primary key default gen_random_uuid(),
  profile_id uuid not null references public.profiles(id) on delete cascade,
  counterparty text not null,
  intent markdown,
  outcome markdown,
  occurred_at timestamptz default now(),
  created_at timestamptz default now(),
  updated_at timestamptz default now()
);

-- 5.7 Rituals & Practices
create table public.rituals (
  id uuid primary key default gen_random_uuid(),

```

```

    profile_id uuid not null references public.profiles(id) on delete cascade,
    title text not null,
    cadence text check (cadence in ('daily','weekly','monthly','custom')) default
'daily',
    instructions markdown,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);

create table public.ritual_sessions (
    id uuid primary key default gen_random_uuid(),
    ritual_id uuid not null references public.rituals(id) on delete cascade,
    did_happen boolean default true,
    notes text,
    occurred_at timestamptz default now(),
    created_at timestamptz default now()
);

-- 5.8 Relationships Graph
create type relation_kind as enum
('mother_child','father_child','sibling_playmate','admired','admiring','partner','friend','collea

create table public.relationships (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid not null references public.profiles(id) on delete cascade,
    person_name text not null,
    kind relation_kind not null,
    started_on date,
    ended_on date,
    notes markdown,
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);

create table public.relationship_events (
    id uuid primary key default gen_random_uuid(),
    relationship_id uuid not null references public.relationships(id) on delete
cascade,
    title text,
    detail markdown,
    occurred_at timestamptz not null,
    created_at timestamptz default now()
);

-- 5.9 Goals (OKR) & Updates
create table public.goals (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid not null references public.profiles(id) on delete cascade,

```

```

    title text not null,
    area_id uuid references public.areas(id),
    dimension_id uuid references public.dimensions(id),
    target_date date,
    status text check (status in ('planned','active','paused','done','dropped'))
default 'planned',
    created_at timestamptz default now(),
    updated_at timestamptz default now()
);

create table public.goal_updates (
    id uuid primary key default gen_random_uuid(),
    goal_id uuid not null references public.goals(id) on delete cascade,
    summary markdown,
    progress numeric,
    created_at timestamptz default now()
);

-- 5.10 Work & Finance Integration (monthly/quarterly)
create table public.work_finance_integrations (
    id uuid primary key default gen_random_uuid(),
    profile_id uuid not null references public.profiles(id) on delete cascade,
    framework_version text default 'v5',
    created_at timesta

```