

SAED YOUNIS

Backend Engineer — Distributed Systems, Payments & Multi-Tenant SaaS

Istanbul, Türkiye · +90 542 102 9945 · Saedbusiness2021@gmail.com · linkedin.com/in/saedyo45 · github.com/Saexd · saedcv2.netlify.app

Work authorisation — Turkish citizen. Eligible for the **EU Blue Card** (Germany IT shortage-occupation route), **Ireland Critical Skills Employment Permit** and the **Netherlands Highly Skilled Migrant** scheme. Degree recognised in Germany's ANABIN database (status H+). Available to relocate at short notice.

SUMMARY

Backend engineer with four years building production APIs, real-time services and payment-critical systems. Architect of a **multi-tenant B2B SaaS platform** on PHP 8.3 / Laravel 13 — 27 bounded domains, 118 migrations, 21 versioned REST modules, an embedding-based ranking pipeline, and a CI gate enforcing PHPStan level 8 with no baseline plus a dedicated tenant-isolation test matrix over 1,570 tests. Has twice designed and shipped the same hard pattern: a **Node.js/Socket.IO real-time tier fronting an authoritative Laravel API**, with server-side authorisation, presence, moderation and rate limiting. On a live crypto & fiat wallet, implemented idempotency across 15+ money-movement flows and canonical SHA-256 intent hashing against replay and tampering. Also ships **async Python** — SQLAlchemy 2.0 over asyncpg and PostgreSQL, Alembic-migrated, Docker-composed. Correctness-first: isolation proven by test, no existence oracles in auth, privacy enforced in the query layer rather than the UI.

Core stack: PHP 8.3 · Laravel 13 · Node.js · TypeScript · Python 3.12 · PostgreSQL / MySQL · Redis · REST · WebSockets · Stripe · AWS · Docker · GitHub Actions / GitLab CI

WORK EXPERIENCE

Founder & Lead Backend Engineer

06/2026 – Present

TruAlly — B2B event matchmaking SaaS · yourtrually.com

Multi-tenant platform in production. Laravel API + Next.js web + Flutter client; sole owner of the backend architecture.

- Architected a **domain-driven Laravel 13 / PHP 8.3 API** — 27 bounded domains (Identity, Tenancy, Billing, Commerce, Matching, Meetings, Messaging, Privacy, Idempotency, Webhooks, Realtime), 118 migrations and 21 versioned REST route modules behind Sanctum token auth.
- Made **tenant isolation an enforced invariant, not a convention**: a query-layer bypass guard with an explicit allowlist, tenant-scoped model binding, and a dedicated CI suite — the **tenant-isolation matrix** — that fails the build on any cross-tenant read.
- Designed the **commerce and billing layer** as auditable state machines (draft → paid → refunded), with plan quotas that hard-block orders at the limit, a **Stripe webhook incident lifecycle** (received → open → retrying → recovered) with retry and operator deep-links, and dunning-driven tenant suspension.
- Engineered the **matching engine** — a weighted multi-feature scorer over embeddings of four row families, explanations generated strictly from the stored feature snapshot so no reason is invented. Diagnosed a production failure serving an empty feed to **2,008 of 2,275 participants**: root cause was operational (ranking runs never executed for 45 of 46 editions), not algorithmic. Shipped debounced re-rank queueing on pool change plus score-threshold recalibration.
- Built the **async pipeline** on queued jobs and cron workers — a profile change fans out to per-tenant projection jobs that re-extract fingerprints and re-queue the embed→rank chain, so an edit genuinely re-ranks instead of going stale.
- Hardened **auth and privacy**: passwordless magic-link with a constant 202 response (no account-existence oracle), hash_equals token comparison, throttled endpoints, a rotatable unauthenticated ICS feed validated against an independent RFC 5545 parser, GDPR anonymisation, and **k=5 cohort suppression** so analytics cannot re-identify small groups. Closed a real leak where private feedback escaped an unconstrained eager-load — proven by injecting the exact leak, then blocking it.
- Found and fixed a class of **client-only authorisation**: four features gated in the frontend alone let a low-tier token read ranked recommendations. Moved enforcement server-side, made an unknown entitlement key **fail closed** instead of open, and added a tenant-wide budget cap so one caller could not drain the AI quota for everyone.
- Owned the **CI/CD security gate** — Pint, **Larastan level 8 with no baseline**, Gitleaks secret scanning, a content-security pattern check and Pest suites (**1,570 tests**) — and ran production deploys.

Tech: PHP 8.3, Laravel 13, Sanctum, MySQL, Redis, Laravel Reverb (WebSockets), queued jobs & cron workers, Stripe, AWS S3, Pest/PHPUnit, PHPStan/Larastan L8, GitHub Actions, Python ETL, Next.js BFF.

Payments & Platform Engineer

01/2026 – Present

Better-Pro Enterprises Limited — HoyaPay · Remote (US)

Full-time 01/2026 – 05/2026; project-based collaboration 06/2026 – Present. HoyaPay: crypto & fiat wallet live on the App Store and Google Play.

- Owned the **money-movement integrity protocol** across 15+ transactional flows — deposit, withdrawal, transfer, exchange, buy/sell, virtual and physical cards — implementing **idempotency keys** that make duplicate submissions safe under retry and network partition.
- Implemented **intent hashing**: a canonical SHA-256 over amount, recipient, currency and path, binding each transaction from prompt to execution so a mutated payload fails verification server-side.
- Designed the transaction-integrity model with the backend team on a strictly **server-authoritative** contract — the client displays and forwards, never decides — with payload-freeze snapshots, double-submit guards and crash-recoverable lifecycle tracking for in-flight transactions.
- Integrated **passwordless authentication** end to end: passkey/WebAuthn (FIDO2) registration and login with OTP fallback and **step-up re-authentication** for sensitive operations, orchestrated through a single verification service with per-device credential management.
- Consumed the **real-time transaction layer** over WebSockets with connectivity-aware reconnection; hardened transport with a typed error taxonomy, encrypted requests, **request signing** and strict logging hygiene — no tokens, balances or PII in any log sink.

Tech: REST, WebSockets, WebAuthn/FIDO2, SHA-256 intent hashing, idempotency keys, request signing, Keychain/Keystore, GitLab CI/CD, Flutter/Dart client.

Bubbles — location-based social platform*Four-codebase platform architected and led end to end; iOS/Android live on TestFlight.*

- Designed and built the **Laravel REST API that owns the truth — 44 models, 74 migrations, ~249 endpoints** behind Sanctum — with multi-provider auth (Apple / Google / phone), FCM push and deep-link resolution.
- Built a **Node.js Socket.IO real-time server** that verifies every socket identity against Laravel before fan-out — **no client-supplied identity is ever trusted** — over a shared-secret internal bridge, replaying room membership on reconnect while preserving nested join order.
- Implemented the real-time modules: presence and rosters, rooms and membership, delivery/read acks, friends and blocks, **moderation with user and IP bans**, a per-socket **rate limiter**, structured logging and Sentry.
- Built **server-authoritative geofencing** — GPS only decides when to ask; the server decides inside/outside via Haversine and is the single source of truth, with transient network errors never ejecting a user, plus location-integrity and anti-bot guards.
- Designed a **presigned-upload media contract** (presign → PUT → complete) to S3 with multipart fallback: messages carry only a media ID and private objects are served through short-TTL signed URLs, so no media bytes transit the chat pipeline.

Tech: Laravel 12, Sanctum, MySQL, Node.js, Socket.IO, Express, AWS S3, FCM (HTTP v1), Sentry, Next.js 16 admin, Docker.**FREELANCE & CONTRACT PROJECTS****Client engagements delivered as contract work**

01/2022 – 12/2025

- AqarClick** (06 – 08/2025) — real-estate platform. Built the **Laravel 12 / PHP 8.2 REST API (100 migrations)**, layered into Services, Policies, Observers and Traits, Sanctum auth, policy-driven RBAC) plus a companion **Node.js Socket.IO service** on a strictly **Laravel-authoritative** contract — the API persists, then pushes to an internal emit endpoint and the socket layer fans out; clients emit only ephemeral signals (typing, presence, read acks), never state. Added socket auth handshakes, room membership, presence and a **per-connection rate limiter** with structured logging and Sentry.
- Teacher-student learning platform** (09 – 12/2025) — Flutter client on Clean Architecture with BLoC against a token-authenticated API: content management, payment flows, role-based access, Firebase (Auth, Cloud Messaging, Analytics) and Arabic RTL localisation.
- Rest Fort** (10/2023 – 01/2024) — integrated multiple **payment gateways** and third-party SDKs; engineered real-time chat over the Pusher SDK with delivery and read state (Bloc + MVVM).
- Masar** (06 – 07/2023) — React admin panel and a mobile client against **Laravel APIs**, keeping web and mobile synchronised on one data contract. **Yumgott** (04 – 05/2023) — **Swish payment** integration and digital-wallet features. **Nabda** (01 – 02/2022) — server-side form submission and validation.

Tech: PHP 8.2, Laravel 12, Sanctum, MySQL, Node.js, Socket.IO, Express, Sentry, REST, RBAC, Pusher, payment gateways, Flutter/Dart, Firebase, React/TypeScript.**SELECTED ENGINEERING PROJECTS****AI Job Hunter — autonomous recruitment agent**

2026

Python 3.12 · async SQLAlchemy · PostgreSQL 16 · Docker Compose

- Containerised **async Python service** that continuously ingests listings, scores them with an LLM, auto-applies to strong matches and reports through a Telegram bot and dashboard.
- Fully asynchronous data layer** — SQLAlchemy 2.0 async ORM over asyncpg against PostgreSQL 16, with **Alembic migrations** run as a dedicated compose service so schema changes deploy ahead of the app; typed config and validation end-to-end with **Pydantic v2**.
- Resilient ingestion via httpx with **tenacity retry policies** and **aiolimiter** concurrency rate limiting so upstream sources are never hammered; recurring pipelines scheduled with APScheduler.

Tech: Python 3.12, SQLAlchemy 2.0 (asyncio), asyncpg, PostgreSQL 16, Alembic, Pydantic v2, httpx, tenacity, aiolimiter, APScheduler, OpenAI API, Playwright, Docker Compose.**Usta Bey — field-service quoting & invoicing platform**

2026

Laravel 13 / PHP 8.3 API · offline-first sync · Flutter client

- Built a **Laravel 13 REST API** with Sanctum auth backing an **offline-first sync engine** — the client queues mutations locally and reconciles against the server, which stays authoritative on conflict.
- Implemented **server-driven session invalidation** and one canonical billing identity, so entitlement state cannot diverge between device and server or leak between accounts; **RevenueCat** entitlements gate Pro document generation.

Tech: PHP 8.3, Laravel 13, Sanctum, REST sync protocol, RevenueCat, Flutter/Dart, i18n (AR/TR/EN + RTL).**EDUCATION****B.Sc. Computer Systems Engineering**

09/2021 – 07/2025

Al-Azhar University — Gaza · Faculty of Engineering & Information Technology

- Cumulative average **87.12%** — **"Very Good"**; 162/162 credit hours. Recognised institution in the German ANABIN database (status **H+**).
- Relevant coursework: **Distributed & Cloud Computing**, Operating Systems (90), Computer Networks (90), **Information Security** (90), Data Structures (95), Compilers Design, Data Mining & Data Science, Advanced Software Engineering (97), Advanced Software Design (97), Graduation Project I & II (98/98).

SKILLS

Languages	PHP 8.3 (expert) · JavaScript / Node.js · TypeScript · Python 3.12 (async, typed) · SQL · Dart
Backend	Laravel 13 · Express / Socket.IO · SQLAlchemy 2.0 (asyncio) · Pydantic v2 · domain-driven design · REST API design & versioning · WebSockets & broadcasting (Reverb, Pusher) · queued jobs, workers & cron pipelines · async concurrency, retries & backpressure · caching
Data	PostgreSQL · MySQL · Redis · schema design & migrations (Laravel + Alembic, 300+ in production) · query optimisation & eager-load control · vector embeddings & ranking pipelines · ETL & idempotent seeding

Architecture	Multi-tenancy & tenant isolation · idempotency & exactly-once semantics · state machines · event-driven & async fan-out · server-authoritative real-time · BFF pattern · Clean Architecture · SOLID
Payments & Security	Stripe (billing, webhooks, dunning) · payment gateways · intent hashing (SHA-256) · request signing · OAuth2 / token auth (Sanctum, JWT) · passkeys / WebAuthn / FIDO2 · RBAC & entitlements · rate limiting & throttling · GDPR & data privacy · OWASP practices
Platform	Docker & Compose · GitHub Actions · GitLab CI/CD · AWS (S3, EC2) · Linux · Nginx · static analysis (PHPStan/Larastan L8) · secret scanning (Gitleaks) · Sentry & observability · Pest / PHPUnit / node:test · Git, Postman
Also	Next.js / React · Flutter (4 yrs, production fintech) · AI-assisted development (Claude Code, Cursor) · Agile / Scrum

LANGUAGES

Arabic (Native) · English (Fluent, C1) · Turkish (Fluent, C1)