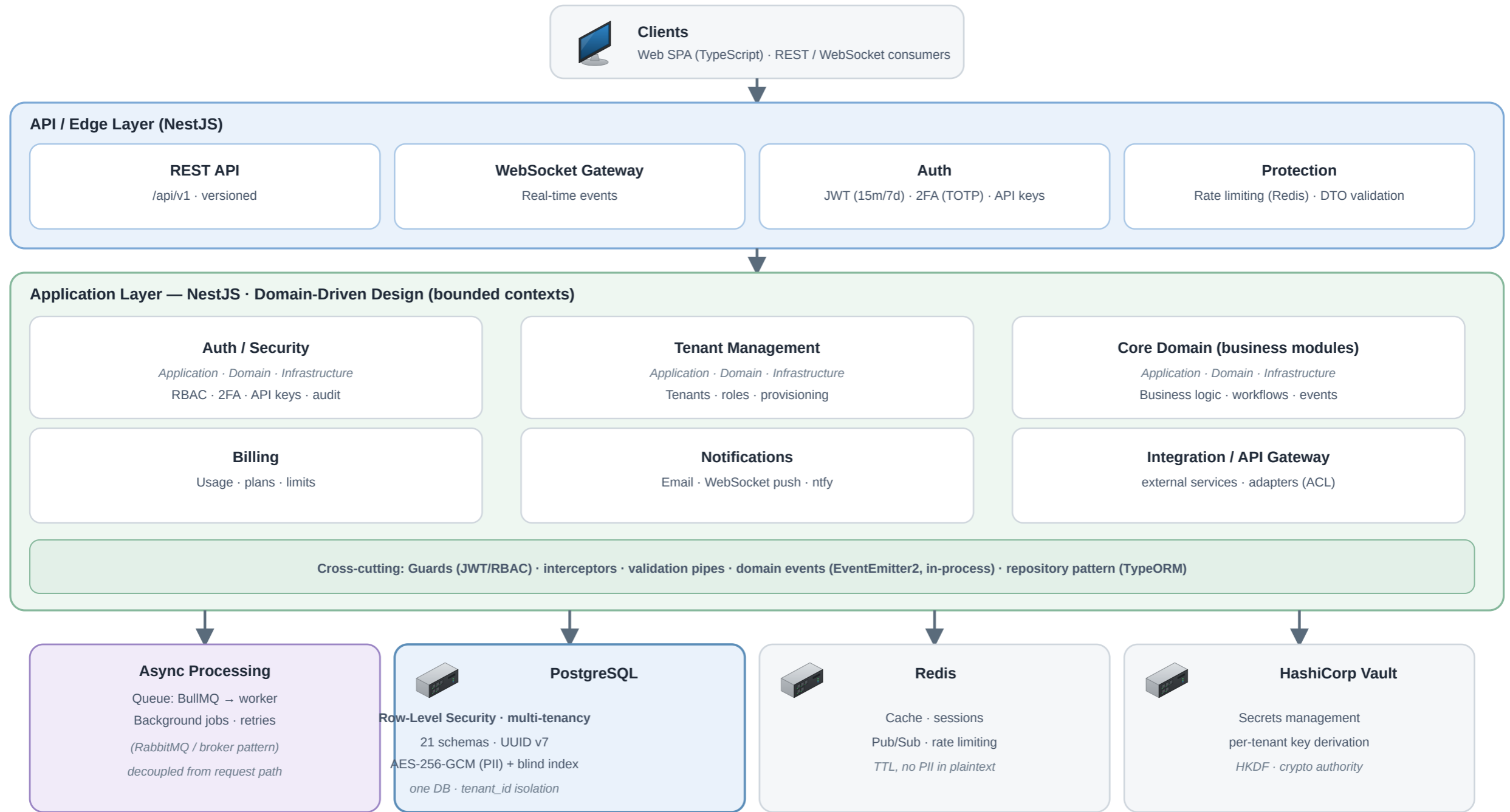


Multi-Tenant SaaS · Software / Backend Architecture

NestJS (Domain-Driven Design) · PostgreSQL Row-Level Security · REST + WebSocket · BullMQ worker · Redis · JWT/2FA · anonymized



★ Tenant isolation (the core of multi-tenancy)

Every request carries a tenant_id → PostgreSQL Row-Level Security (FORCE) enforces per-tenant row isolation — one DB, hard separation.

Per-tenant PII encrypted with AES-256-GCM, keys derived via Vault (HKDF). The domain layer stays framework-free (DDD); external SDKs only via adapters (ACL).

 Data stores (PostgreSQL · Redis · Vault)

 Client (web SPA / API consumer)

 Edge/API + database

 Application Layer (DDD)

 Async (queue/worker)

Anonymized architecture pattern of my own multi-tenant SaaS platform — generic bounded contexts, no product internals. Stack 1:1 as operated.

Stack: NestJS · PostgreSQL/RLS · Redis · BullMQ · WebSocket · JWT/2FA · TypeORM · DDD · anonymized

