

Zandaris Backend – Architecture

System Overview

Zandaris is an internal police information system built for **Mestská polícia Bratislava** (Municipal Police of Bratislava). It manages the full lifecycle of police events (udalosti) – traffic violations, parking offenses, towing, wheel clamp placement, and related administrative processes including penalty ticket management, officer settlements, and statistical reporting.

The backend is a **NestJS** application backed by **PostgreSQL** (via Prisma ORM), **MinIO** for file storage, and **Azure AD** for authentication. It integrates with several Slovak government SOAP/REST services for vehicle and person lookups.

Environments

Environment	URL
Production	https://zandaris.bratislava.sk/
Staging	https://zandaris.staging.bratislava.sk/
Development	https://zandaris.dev.bratislava.sk/

High-Level Architecture

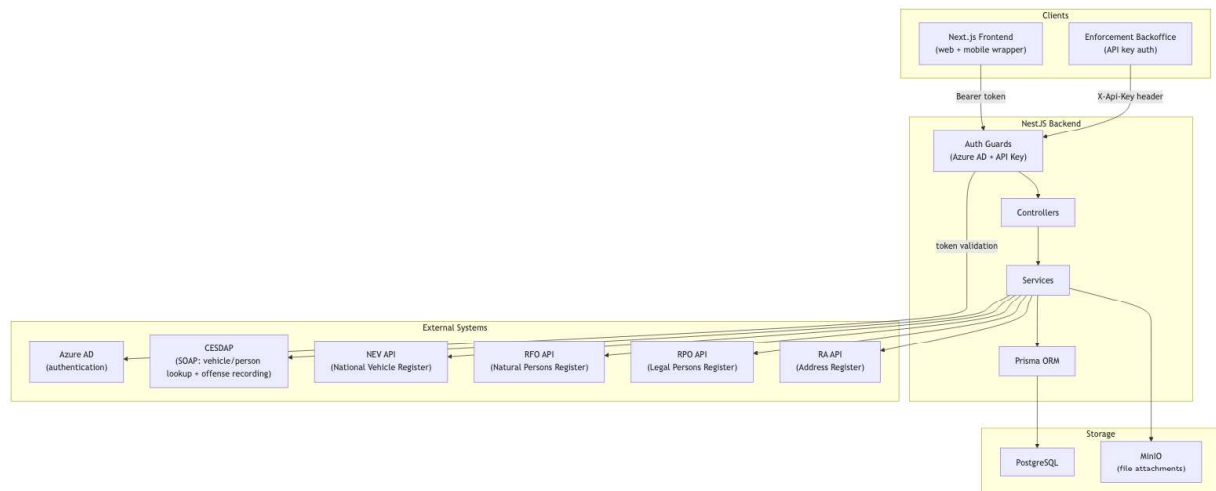


Figure 1: Diagram

Module Map

The application is organized into the following NestJS modules, all imported by the root AppModule:

Module	Directory	Purpose	README
AppModule	<code>src /</code>	Root module, global config, static file serving	–

Module	Directory	Purpose	README
AuthModule	src/auth-aad/	Azure AD + API key authentication, role guards	README
PrismaModule	src/prisma/	Global database access (Prisma client)	README
EventsModule	src/events/	Core: events, tickets, batches, eliminations, stats	README
IndexesModule	src/indexes/	Reference data CRUD (20+ entity types)	README
HomeModule	src/home/	Dashboard, patrol tracking, street data	README
InfoModule	src/info/	User profile, substitutions, diagnostic messages	README
BackofficeModule	src/enforcement-backoffice/	Integration with enforcement backoffice system	README

External API client code (auto-generated from OpenAPI specs) lives in src/ext/:

Client	Directory	Service
BO	src/ext/bo/	Enforcement Backoffice API
EVO	src/ext/evo/	National Vehicle Register (Národná evidencia vozidiel)
RA	src/ext/ra/	Address Register (Register adries)
RFO	src/ext/rfo/	Natural Persons Register (Register fyzických osôb)
RPO	src/ext/rpo/	Legal Persons Register (Register právnických osôb)

See also: [External API Clients README](#)

Data Model Overview

The central entity is **Event** (police event/incident). All other entities either classify events, track their lifecycle, or support organizational structure.

Key Domain Concepts

- **Event**: A police action (violation, towing, blocker placement, etc.). Contains embedded vehicle, person, and legal person data enriched from government registers.
- **EventType** / **EventGroup**: Hierarchical classification tied to specific laws and paragraphs.
- **Resolution**: How an event was concluded (fine, warning, forwarded to court, etc.).

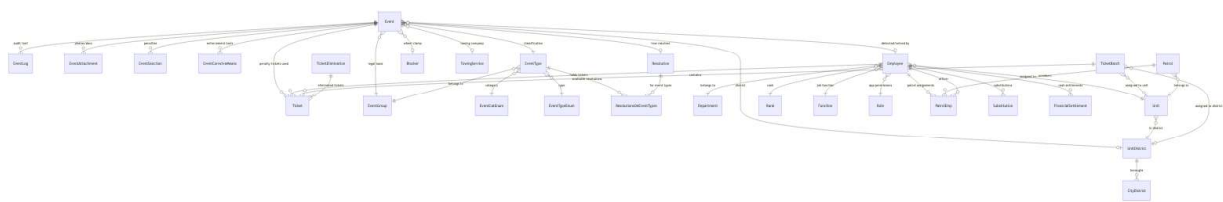


Figure 2: Diagram

- **Ticket (pokutový blok):** Physical penalty ticket booklet. Managed in batches, distributed to units, then to officers.
- **Blocker (papuča):** Wheel clamp device tracked by unit.
- **Patrol:** Named group of officers on shift, with real-time GPS tracking.
- **Employee:** Officer with role-based access. Linked to Azure AD for authentication.

Index Tables

Many entities serve as “index tables” (číselníky) – lookup/reference data managed via the Indexes module. These include: VehicleColor, VehicleBrand, VehicleStatus, VehicleType, LicensePlateType, Country, Detection, DetectionGroup, CoerciveMean, Sanction, Department, Rank, Function, DocumentType, TowingService, PosTerminal, Blocker, Unit, UnitDistrict, Role, Patrol, EventGroup, EventType, Resolution.

All index tables follow a common pattern: soft-delete via `validUntil`, audit fields (`createdBy`, `modifiedBy`, `createdAt`, `modifiedAt`), and validity window (`validFrom`, `validUntil`).

Authentication & Authorization

Authentication Flow

1. The Next.js frontend authenticates users via **NextAuth** with an **Azure AD** provider.
2. Azure AD returns a JWT access token stored in the session.
3. Every API request includes the token as `Authorization: Bearer <token>`.
4. The `AzureAdGuard` validates the token against Azure AD’s public keys and extracts user identity.
5. The `AzureADStrategy` (Passport) loads the corresponding `Employee` record from the database by email.

For machine-to-machine communication (enforcement backoffice), an **API key** is passed via the `X-API-Key` header and validated by the `ApiKeyGuard`.

Routes decorated with `@Public()` skip authentication entirely (used for health checks, street lookups).

Role-Based Authorization

The `RolesGuard` checks whether the authenticated user’s role matches the required roles for each endpoint. Roles are defined in `src/auth-aad/roles.config.ts` and applied via the `@Roles()` decorator.

Role Hierarchy (from most to least privileged):

Role	Description
SYSTEMADMIN	Full system access
CHIEF	Chief officer – read all, statistics, ticket management

Role	Description
CHIEFLOCAL	Local chief – same as CHIEF but scoped to their unit
ADMIN / ADMINMAP	Administrative access (index management)
SHIFTCMDR / SHIFTCMDRMAP	Shift commander – event creation and management
SOLVER / SOLVERMAP	Event solver – handles assigned events
PENALTYBLOCK	Penalty blocker operator
MOBILEPATROL	Mobile patrol officer
TICKETADM	Ticket administration (central)
TICKETLOCAL	Ticket administration (local/district)
MAPONLY	Map-only access

Roles are combined into permission groups: eventRead, eventWrite, eventCreate, indexRead, indexWrite, ticketRead, ticketWrite, statRead, mapsRead, sysadmin, all, etc.

External Integrations

CESDAP (SOAP)

Two SOAP services connected via nestjs-soap:

- **CESDAP__LUST**: Vehicle and person lookup (lustrácia). Used when officers need to check a license plate or person ID against central records.
- **CESDAP__ZAP**: Offense recording (zápis). Used to register resolved events in the central system.

WSDL files are bundled in src/wSDL/ and loaded at startup.

Government Register APIs (REST)

Auto-generated OpenAPI clients in src/ext/:

- **EVO (NEV)**: National Vehicle Register – vehicle ownership, registration status, technical data.
- **RFO**: Natural Persons Register – person identity, address, status.
- **RPO**: Legal Persons Register – company data by IČO.
- **RA**: Address Register – address validation and geocoding.

MinIO (Object Storage)

File attachments (event photos, ticket elimination documents) are stored in MinIO buckets. Legacy attachments were stored as Bytes in PostgreSQL and are being migrated to MinIO via the migrateattachmentstobucket sysadmin endpoint.

Enforcement Backoffice

Events can be forwarded to an external enforcement backoffice system via REST API (BackofficeModule). This system handles the formal legal proceedings that follow a police event. Communication is authenticated via API key.

Request Lifecycle

A typical authenticated API request flows through the following stages:

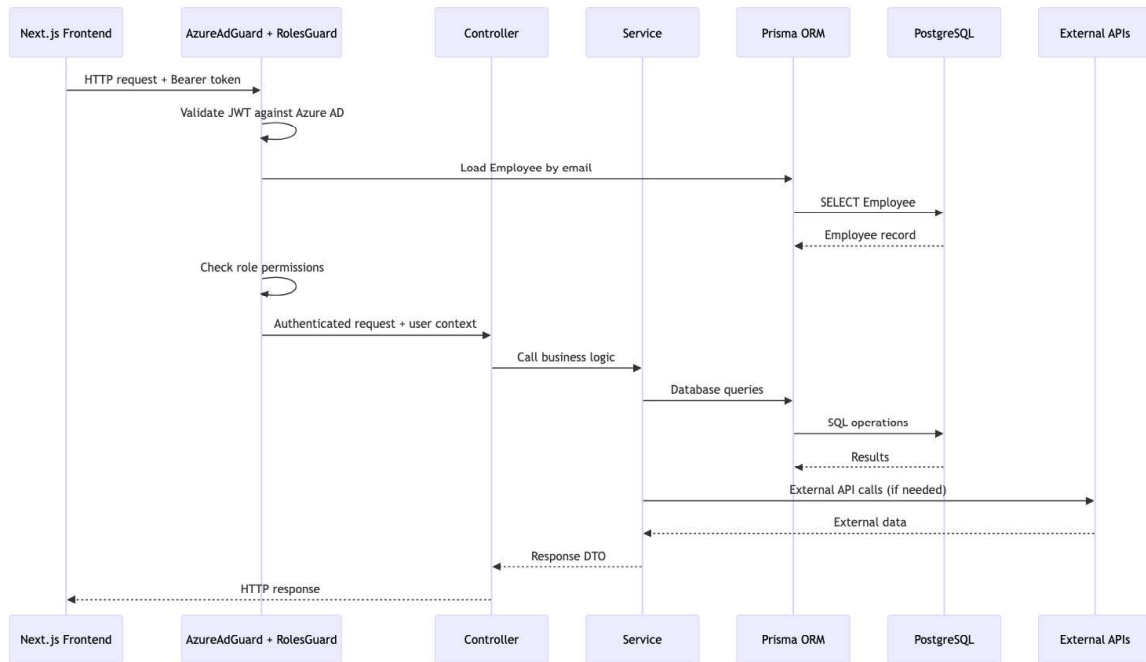


Figure 3: Diagram

Key middleware in the pipeline: - **ValidationPipe** (global): Validates request DTOs using class—validator decorators. - **PrismaExceptionsFilter**: Catches Prisma errors and converts them to user-friendly HTTP exceptions with Slovak error messages. - **FileInterceptor**: Handles multipart file uploads for event attachments.

Deployment

The application runs on **Kubernetes** with environment-specific configurations in `kubernetes/envs/`:

- Dev/ – development cluster
- Staging/ – pre-production testing
- Prod/ – production

Database Migrations

Prisma migrations are stored in `prisma/migrations/`. Key deployment notes:

1. Test migrations against a copy of production data on staging before merging to master.
2. Coordinate deployment windows with police staff (minimum 2-hour window).
3. Manual database backup before production deployment via [db-backup-k8s](#) pipeline.
4. Production releases use tags prefixed with prod.

See the [project README](#) for the full deployment checklist.

Swagger / OpenAPI

Interactive API documentation is available at the `/api` endpoint on any running instance. The OpenAPI spec is also available as JSON at `/spec-json`.

The frontend generates its TypeScript API client from this spec (see `_parking-police/zandaris/next/src/services/client`).