

SOFTWARE WHITEPAPER • EDITION 1.0

Digital fire department administration as a secure platform

Architecture, client separation, data model, security principles and operating model of IGNIS 2AB - developed for municipal fire brigade structures and comprehensible administrative processes.

Executive summary

IGNIS 2AB bundles personnel, training, operations, vehicles, equipment, warehouses, organization, documents and selected financial processes in a web-based platform. The system maps a main fire department with subordinate local fire departments without breaking down local areas of responsibility.

The architecture combines a multi-tenant Symfony application, PostgreSQL, private file storage, role and fire department authorization, and a separate licensing service. Development, demo and production are organizationally and technically separated.

1. Initial situation and target image

Fire department administrations often work with heterogeneous specialist applications, tables, paper files and local filings. This creates media disruptions, double maintenance, inconsistent authorizations and low transparency about qualifications, operational readiness and technology.

IGNIS 2AB pursues a modular goal: a common database, unique organizational identifiers, fine-grained rights, verifiable life cycles and standardized exports - while still clearly separating the individual fire departments.

2. Technical architecture

Application

Symfony 7.4, PHP 8.2+, Twig, server-side validation and security components.

database

PostgreSQL with functional schemas, UUIDs and additional client separation.

Operation

Containerized operation with Nginx, reproducible deployments and health checks.

files

Private upload storage outside of the public webroot, UUID filenames and checksums.

e-mail

Authenticated SMTP sending with TLS for verification, reporting and notifications.

Licensing

Separate application independent service on license.2ab.it with signed API.

Browser/client



TLS/Reverse Proxy



IGNIS 2AB



PostgreSQL + Private Files



License and mail services

3. Organizational and client model

Each customer organization has an immutable Org ID. Main and local fire departments receive site IDs; each deployed instance has an installation ID. These UUID-based identifiers prevent name-based ambiguity while providing the basis for interfaces and licensing.

Technical data is assigned to a client and – if technically necessary – to a fire department. Local fire departments generally see and process their own data. Central roles only receive the expressly intended cross-organizational reading or administration rights.

The application combines repository filters, permission checks and database mechanisms. The active fire department is an explicit user context, not just surface filtering.

4. Specialist modules

staff

Master data, ranks, functions, groups, qualifications, documents and organizational charts.

Training

Catalogues, bookings, examination, approval, rejection and personal training files.

Mission

Operations, participants, vehicles, equipment, material, timeline, reports and certificates.

Technology

Vehicles, loading, devices, radio technology, testing, maintenance and lifecycle events.

Warehouse

Items, inventory, receipts, expenses, inventory and orders.

organization

Fire department structure, user mapping, logos, licenses and import tools.

Finance

Selected processes such as employer-related loss of earnings with protected access.

Documents

Uniform connection with people, vehicles, devices and operations.

5. Security architecture

Identity and access

Individual user accounts, password hashing, login throttling, CSRF protection, secure sessions and two-factor authentication protect access. Privileged roles should use 2FA mandatorily.

Authorization

Roles, client membership, active fire department and property-specific voters are evaluated together. Unauthorized objects can intentionally appear as non-existent to prevent information leakage.

Transport and API

Public connections use TLS. APIs require tokens without exception. License checks are protected against manipulation and replay with HMAC-SHA256, timestamp and nonce; Answers are also signed.

Infrastructure

Host firewall, fail2ban, key-based SSH access, separate accounts, security headers and controlled deployments complement application controls.

6. Data and document model

UUIDs serve as stable technical primary keys. Technical identifiers such as personnel numbers remain separate. Life cycle models document inspections, maintenance, expenses, returns, retirements, entries, exits, promotions and changes in function.

Uploads are stored privately, provided with internal file names and SHA-256 checksums and delivered via controlled download routes. Document links allow subject-specific storage without public file paths.

Information that is particularly sensitive to protection, such as suitability or loss of earnings, requires restricted roles, auditing and set deletion periods.

7. Migration and integration

Imports from alternative systems process supplied backups through a controlled flow of upload, technical validation, preview and expressly approved import. Source IDs, checksums, target objects and warnings are logged.

Imports do not create a tacit technical truth: assignments, duplicates, number ranges and organizational structures must be technically checked and approved. A structured export supports data portability and end of contract.

Public specialist APIs are only activated after versioning, rights concepts, data minimization and documented release. The current API documentation only describes the existing license and provisioning API.

8. License and operating model

A client license covers the central fire department. Local fire departments are managed as independently licensable stations under the organization. The license server only stores necessary organizational and installation metadata and does not store fire department personnel data.

Active and trial licenses provide write access; expired, suspended, revoked, or unlicensed units are placed in read mode. License status is checked regularly and can be safely cached within a limited offline grace.

9. Development, quality and operations

The development server is the single source of truth. Changes are versioned, tested and initially checked in a demo environment. Production is consciously and updated with database backup, migration check, rollback plan and functionality check.

Health checks, application and system logs, storage and RAID monitoring as well as backup and restore procedures support operations. Build identifiers differentiate development, demo and production.

In addition to technical logic, tests focus particularly on authorizations, client separation, organizational changes, import consistency, license behavior and security-critical processes.

10. Data protection and governance

Municipalities and customers remain responsible for their specialist data; 2AB processes them within the framework of a contract in accordance with Art. 28 GDPR. Processing plant, TOM, subcontractor list and instruction form specify the responsibilities.

Data protection by design includes data minimization, private storage, privilege separation, logging, export, rectification and deletion processes, and development data separation. Deletion periods are bindingly decided and technically implemented for each data category.

A data protection impact assessment, the list of processing activities and acceptance by data protection officers are part of the organizational production release.

11. Differentiation and further development

IGNIS 2AB supports administrative and documentation processes, but does not replace operations management, medical assessment, legal review or legally required professional decisions.

The platform is being further developed in a modular manner. Future focuses will be standardized specialist APIs, further data protection automation, malware checking of uploads, stricter content security policies, expansion of automated security checks and additional integrations.