GDPR Compliance Report - Sprint C

Port Management System

LEI - SEM5 - PI 25/26

TURMA 3DE

GRUPO 05

Date: January, 2026
Project: Port Visit Notification and Management System
Sprint: Sprint C - Operations & Execution Management (OEM) Module
Previous Versions:

- GDPR_Compliance_Report_SprintA.pdf
- GDPR_Compliance_Report_SprintB.pdf

Executive Summary

This report provides a comprehensive GDPR compliance analysis for the Port Management System at the conclusion of Sprint C. Building upon the foundations established in Sprint A (core domain model) and Sprint B (authentication and authorization), Sprint C introduces the **Operations & Execution Management (OEM) module** - a decoupled Node.js/TypeScript backend service that manages the execution phase of port operations.

Sprint C represents a significant architectural evolution: the system now operates as a **distributed microservices architecture** with two independent backends (Core Backend in .NET 9 and OEM Backend in Node.js) and introduces a **new data storage technology** (MongoDB) alongside the existing SQL Server database. This architectural shift creates new GDPR considerations around data consistency, cross-service data flows, and distributed audit logging.

Key Sprint C Additions:

- **OEM Backend Module** (Node.js + TypeScript + MongoDB) - US 4.1.1
- **Operation Plan Management** - US 4.1.2, 4.1.3, 4.1.4, 4.1.5, 4.1.6
- **Vessel Visit Execution Tracking** - US 4.1.7, 4.1.8, 4.1.9, 4.1.10, 4.1.11
- **Incident Management** - US 4.1.12, 4.1.13
- **Complementary Task Management** - US 4.1.14, 4.1.15
- **Integration with Planning Algorithms** (Prolog-based scheduling)
- **RESTful Inter-Service Communication** (no direct database access)
- **GDPR Compliance Features** (Core Backend):
  - **Privacy Policy Management** - US 4.5.1 (Display), US 4.5.2 (Acknowledgment/Consent)
  - **User Data Rights (SAR)** - US 4.5.3 (Export, Rectification, Deletion)
  - **Non-User Data Requests** - US 4.5.4 (External data subject requests)

Overall GDPR Compliance Status for Sprint C:

- ☑ **Strengths:** User ID references only (no duplication), audit trail in operation plans, MongoDB supports field-level encryption
- ☑ **NEW - Privacy Policy System:** Full privacy policy management with versioning and user acknowledgment (US 4.5.1, US 4.5.2)
- ☑ **NEW - Subject Access Request (SAR):** Users can export, request rectification, and request deletion of their data (US 4.5.3)
- ☑ **NEW - Non-User Data Requests:** External data subjects can submit GDPR requests via public form (US 4.5.4)
- ⚠ **Moderate Gaps:** User data fetched from Core Backend (dependency), incident data may involve external entities, no MongoDB encryption configured yet
- ⚠ **Remaining Gaps:** OEM Backend integration for complete data export, email notifications for data requests, no DPA with third parties

Sprint C-Specific GDPR Concerns (Partially Addressed):

1. **Distributed Personal Data:** User IDs stored in OEM (MongoDB) reference personal data in Core Backend (SQL Server)
2. **External Entity Involvement:** Incidents may involve third parties (Coast Guard, Fire Department) - potential data sharing
3. **Dual Database Architecture:** Personal data split across SQL Server and MongoDB requires coordinated retention/erasure
4. **Cross-Service Audit Trails:** Actions span multiple systems, making comprehensive audit logging challenging
5. **No Data Processing Agreement:** Core Backend and OEM Backend treated as separate systems without formal DPA

Sprint C GDPR Improvements Implemented:

1. ☑ **Privacy Policy System (US 4.5.1, 4.5.2):** Public privacy policy display, version management, user consent tracking
2. ☑ **Subject Access Request Mechanism (US 4.5.3):** Data export (JSON/PDF), rectification requests, deletion requests with soft-delete
3. ☑ **Non-User Data Rights (US 4.5.4):** Public form for external data subjects to exercise GDPR rights
4. ☑ **Data Request Tracking:** Status management (Pending → InProgress → Completed/Rejected) with admin oversight

---

1. Current Implementation - Personal Data Inventory

1.1. Personal Data in Sprint C (OEM Module)

The OEM module **does not directly store traditional personal data** (names, emails, national IDs). Instead, it stores **user identifiers** that reference personal data maintained in the Core Backend. This design follows microservices best practices but creates GDPR implications for distributed data management.

A. User Identifiers in OEM Module

Stored in MongoDB Collections:

1. **OperationPlan Collection:**
    - `createdBy` (string - User ID from Core Backend)
    - `auditLog[].userId` (string - User ID for change tracking)
    - `auditLog[].userName` (string - **PERSONAL DATA** - cached for convenience)
2. **VesselVisitExecution Collection:**
    - `completedBy` (string - User ID)
    - `operations[].staffAssigned[]` (array of Staff IDs from Core Backend)
3. **Incident Collection:**
    - `reportedBy` (string - User ID or name)
    - `investigatedBy` (string - User ID)
    - `resolvedBy` (string - User ID)
    - `closedBy` (string - User ID)
    - `notes[].author` (string - User ID or name)
    - `externalEntitiesInvolved[]` (array of strings - **THIRD-PARTY ENTITIES**)
4. **ComplementaryTask Collection:**
    - `assignedStaff[]` (array of Staff IDs)
    - `assignedTeam` (string - team name, may include personal identifiers)
    - `completedBy` (string - User ID)

GDPR Classification: Indirect personal data (online identifiers per Article 4(1))

Legal Basis: Same as Core Backend - Contract 6(1)(b) for operational users, Legal Obligation 6(1)(c) for crew-related data

Legal Basis - International Treaty Obligations:

Portugal ratified the **FAL Convention (Convention on Facilitation of International Maritime Traffic, 1965)** via Decreto n.º 13/1990. This international treaty establishes **mandatory data collection requirements** for port authorities, providing strong legal basis under GDPR Article 6(1)(c) - Legal Obligation.

FAL Convention Mandates Collection Of:

- **Crew Lists** (Article 2.6): Name, nationality, rank, date/place of birth, identity document details
- **Passenger Lists** (Article 2.7): Name, nationality, date/place of birth, embarkation/disembarkation ports
- **Health Declarations** (Article 2.9): Health status of persons on board (special category data under GDPR Article 9)
- **Cargo Declarations** (Article 2.3): Details of goods, linked to responsible parties
- **External Entity Coordination** (Article 4): Cooperation with customs, health authorities, coast guard

GDPR Implication: Personal data collected in compliance with FAL Convention has legitimate legal basis under Article 6(1)(c), but **data minimization still applies** - only data explicitly required by the convention should be collected.

Reference: Decreto n.º 13/1990 (Portuguese ratification), FAL Convention Articles 2.6, 2.7, 2.9, 4

Critical Design Decision: OEM stores **only user IDs**, not full user objects. When displaying operation details, the frontend must fetch user details (name, email) from Core Backend API.

GDPR Implication:

- ☑ **Data Minimization:** OEM doesn't duplicate personal data
- ⚠ **Right to Erasure Complexity:** Deleting a user in Core Backend leaves orphaned IDs in OEM
- ⚠ **Transparency:** Users may not be aware their IDs are stored in a separate system
- ❌ **Audit Trail Exception:** `auditLog[].userName` duplicates personal data for performance (violates minimization)

B. External Entity References in Incidents

Field: `Incident.externalEntitiesInvolved[]`

Purpose: Track third-party involvement in operational incidents (e.g., "Coast Guard", "Fire Department", "Customs", "Police")

Data Type: Array of strings (free-text entries)

FAL Convention Legal Basis for External Entity Data Collection:

FAL Convention Article 4: "Contracting Governments undertake to cooperate directly or through the Maritime Organization on matters relating to formalities, documentary requirements, and procedures."

Legal Basis for External Entity Data Collection:

- **Customs Authorities:** FAL Convention Article 2.3 (cargo declarations), Article 3.9 (customs inspection)
- **Health Authorities:** FAL Convention Articles 4.1-4.11 (quarantine, health declarations)
- **Fire Department / Coast Guard:** FAL Convention Articles 5.11-5.12 (emergency assistance)

GDPR Article 6(1)(c) Justification: Recording external entity involvement in incidents is necessary to **comply with FAL Convention Article 4** (international cooperation obligations for port authorities).

GDPR Concerns:

- **Potential Personal Data:** If entry includes individual names (e.g., "Inspector João Silva - Customs"), this becomes personal data
- **Third-Party Data Sharing:** If incident reports are shared with shipping agents, third-party entity names are disclosed
- **No Data Processing Agreement:** No formal DPA with external entities for mutual incident reporting
- **International Transfers:** If external entities are non-EU (e.g., US Coast Guard for international ports), data transfer safeguards required

Recommendation:

- **Document legal basis:** FAL Convention Article 4 provides legal obligation (Article 6(1)(c)) for recording external entity involvement
- Standardize entity names to **official entity designation only** (e.g., "Portuguese Customs Authority", "Fire Department") rather than individual officer names, unless individual identification is required by

the FAL Convention (it is not)
- Add consent mechanism if sharing incident reports with third parties not covered by FAL Convention
- Draft data sharing agreements with regular external entities (Customs, Coast Guard, Health Authorities)

C. VVN and Vessel References (External Data)

The OEM module frequently references **VVN IDs** (Vessel Visit Notification identifiers) and **vessel IMO numbers**. While these are not personal data themselves, they are **linked to personal data** in the Core Backend:

- VVN contains captain CitizenId, crew CitizenId, captain name, crew names
- Vessel registration may include owner information

GDPR Implication:

- OEM operations can be **re-identified** by joining VVN IDs with Core Backend data
- Any OEM data breach could facilitate access to sensitive crew data via VVN linkage
- Article 4(1) pseudonymization: VVN IDs function as pseudonyms for vessels and their crew

---

D. Staff Assignment Data

Fields:

- `VesselVisitExecution.operations[].staffAssigned[]`
- `ComplementaryTask.assignedStaff[]`

Content: Staff mecanographic numbers (employee IDs from Core Backend)

GDPR Classification: Employee data (special protection under labor law + GDPR)

Legal Basis: Contract 6(1)(b) (employment) + Legal Obligation 6(1)(c) (safety regulations require staff assignment tracking)

Sensitive Aspect:

- Staff work schedules can be reconstructed from execution records
- Performance metrics derivable (e.g., operations per staff member, average completion times)
- May reveal health issues indirectly (e.g., staff consistently assigned lighter tasks)

Current Protection:

- ☑ RBAC: Only PortAuthorityOfficer and Admin can assign staff
- ⚠ No access logging: Doesn't track who viewed staff assignments
- ❌ No data retention: Historical staff assignments retained indefinitely

---

E. Health Data in Maritime Operations (Special Category Data)

FAL Convention Requirement: Article 2.9 requires **Maritime Health Declarations** (Declaração marítima de saúde) reporting health status of persons aboard vessels.

GDPR Classification: Article 9 Special Category Data (health data)

Legal Basis: Article 9(2)(i) - "processing is necessary for reasons of public interest in the area of public health, such as protecting against serious cross-border threats to health"

Data Collected:

- Health declaration status for arriving vessels
- Quarantine requirements (FAL Convention Section 4)
- Vaccination certificates (FAL Convention Article 3.7, 4.5)
- Medical emergency incidents (FAL Convention Articles 2.17-2.24)

Current Status in System:

- ⚠ Health declarations not yet implemented in Core Backend (Sprint A-C scope)
- ⚠ Incident module may capture health-related incidents in free-text fields (uncontrolled)

GDPR Concerns:

- ❌ **Special Category Data Risk:** If incident descriptions mention crew member health issues (e.g., "Staff member hospitalized during operation"), this is Article 9 special category data without proper safeguards
- ❌ **No Health Data Processing Controls:** No specific measures for handling health data (encryption, access restrictions)
- ⚠ **FAL Convention Compliance Gap:** Health declarations required by maritime law but not yet implemented

Recommendations:

1. When implementing health declarations (future sprint), ensure special category data protections:
    - Encryption at rest and in transit
    - Role-based access (only health authority and admin)
    - Separate audit logging for health data access
    - Data Processing Impact Assessment (DPIA) required per Article 35
2. **Immediate:** Update incident reporting training to flag health data as requiring extra caution
3. Implement automated detection of health-related keywords in incident descriptions (e.g., "injury", "illness", "hospital", "medical")
4. Create structured health incident fields instead of free-text (e.g., checkboxes for "Medical Emergency", "Quarantine Required")

FAL Convention Requirements for Future Implementation:

- Article 4.1: Health authorities must apply International Health Regulations
- Article 4.5: Vaccination certificate requirements (Yellow Fever, Cholera per WHO)
- Article 4.9: Emergency medical facilities at ports
- Article 2.17-2.24: Medical emergency procedures for vessels

---

1.2. Personal Data Inherited from Sprint A & B (Core Backend)

The OEM module depends on the Core Backend for user authentication and personal data. All personal data categories identified in Sprint A and Sprint B remain applicable:

Quick Reference (Detailed in Sprint A/B Reports):

- **Users:** UserId, Name, Email, ProfilePictureUrl, GoogleUserId, Role, Organization
- **Staff Members:** MecanographicNumber, ShortName, Email, Phone, Qualifications, WorkSchedule
- **Representatives:** Name, CitizenId, Nationality, Email, Phone
- **Crew Members:** Name, CitizenId, Nationality
- **Captains:** Name, CitizenId, Nationality

Sprint C Impact:

- OEM API calls reference these entities via IDs
- JWT tokens from Core Backend carry user claims (name, email, role, orgId)
- OEM validates JWT signature but doesn't verify user still exists in Core Backend (potential stale data)

1.3. Data Processing Activities - Sprint C Additions

Processing Activity	Personal Data Involved	Legal Basis	Data Subjects	Storage Location
Generate Operation Plans	User ID (creator), Staff IDs (assignments)	Legitimate Interest 6(1)(f)	Port staff, operators	MongoDB
Search/List Operation Plans	User ID (creator), Staff IDs	Legitimate Interest 6(1)(f)	Port staff, operators	MongoDB
Update Operation Plans	User ID (editor), User Name (audit), Staff IDs	Legitimate Interest 6(1)(f)	Port staff, operators	MongoDB
Create Vessel Visit Execution	User ID (creator), VVN ID (links to crew)	Legal Obligation 6(1)(c)	Operators, crew (indirect)	MongoDB
Update VVE with Executed Operations	User ID (operator), Staff IDs (workers)	Legal Obligation 6(1)(c)	Port staff, operators	MongoDB
Complete VVE	User ID (completor)	Legal Obligation 6(1)(c)	Operators	MongoDB
Record Incidents	User ID (reporter, investigator), External Entity Names	Legitimate Interest 6(1)(f) + Legal Obligation 6(1)(c)	Port staff, external entities	MongoDB
Manage Incident Types Catalog	User ID (creator)	Legitimate Interest 6(1)(f)	Administrators	MongoDB
Record Complementary Tasks	User ID (assigner, completor), Staff IDs	Contract 6(1)(b) (staff)	Port staff	MongoDB

Processing Activity	Personal Data Involved	Legal Basis	Data Subjects	Storage Location
Query Resource Allocation	Staff IDs (aggregated)	Legitimate Interest 6(1)(f)	Port staff (analytics)	MongoDB
Inter-Service Communication	User IDs, VVN IDs	Same as source data	All subjects	Network transit (HTTPS)

1.4. Third-Party Data Processors and Service Integrations

Processor/Service	Role	Data Processed	DPA Status	Sprint	New in C?
Google OAuth	Authentication provider	Email, name, Google ID	❌ Missing	B	No
Gmail SMTP	Email delivery	Activation emails	❌ Missing	B	No
MongoDB Atlas (if cloud-hosted)	Database hosting	All OEM data (user IDs, incidents, plans)	⚠ Check contract	C	**Yes**
Core Backend .NET	Data source	User personal data via API	❌ No formal DPA	A+C	**Yes**
Prolog Planning Service (future)	Scheduling algorithm	VVN IDs, vessel data (no personal data)	⚠ Future concern	C	**Yes**
Cloud Hosting (Azure/AWS)	Infrastructure	All system data	⚠ Assumed	A	No

Critical Gap - Sprint C: The **Core Backend and OEM Backend** are developed as separate services but lack a formal **Data Processing Agreement** or **Joint Controller Agreement**. From a GDPR perspective:

- **Scenario 1:** If both are controlled by the same port authority → Internal data flows (no DPA needed)
- **Scenario 2:** If OEM is operated by a third-party vendor → OEM is a **processor**, requires Article 28 DPA
- **Scenario 3:** If both jointly determine processing purposes → **Joint controllers** (Article 26), requires joint controller agreement

---

1.5. GDPR Data Subject Rights Implementation (US 4.5.x)

New in Sprint C: The following GDPR features have been implemented in the Core Backend (.NET) to address data subject rights:

A. Privacy Policy Management (US 4.5.1, US 4.5.2)

Domain Layer:

- `PrivacyPolicy` entity - Versioned privacy policy documents
- `PrivacyPolicyAcknowledgment` entity - User consent tracking

Database Tables (SQL Server):

- `PrivacyPolicies` - Stores policy versions with content, effective dates
- `PrivacyPolicyAcknowledgments` - Records user consent with timestamps

API Endpoints:

Method	Endpoint	Description	Auth
GET	`/api/v1/PrivacyPolicy/current`	Get current active policy	Public
GET	`/api/v1/PrivacyPolicy/versions`	List all policy versions	Admin
POST	`/api/v1/PrivacyPolicy`	Create new policy version	Admin
POST	`/api/v1/PrivacyPolicy/acknowledge`	Record user consent	User
GET	`/api/v1/PrivacyPolicy/acknowledgment/status`	Check user's consent status	User

Frontend:

- `/privacy-policy` - Public privacy policy page
- Consent modal on login for new policy versions
- i18n support (EN, PT)

GDPR Articles Addressed:

- Article 7 (Conditions for consent)
- Article 12 (Transparent information)
- Article 13 (Information to be provided)

B. User Data Rights / Subject Access Requests (US 4.5.3)

Domain Layer:

- `DataRequest` entity - Extended with `Source` (USER/EXTERNAL), `UserId`, `RequestData`
- `DataRequestType` enum - ACCESS, RECTIFICATION, ERASURE, PORTABILITY

Application Layer:

- `IDataRightsService` interface
- `DataRightsService` implementation with methods:
  - `ExportUserDataAsync()` - Collects all user data
  - `ExportUserDataAsPdfAsync()` - Generates PDF export
  - `RequestRectificationAsync()` - Creates rectification request
  - `RequestDeletionAsync()` - Creates deletion request
  - `ProcessDeletionRequestAsync()` - Admin processes with soft-delete

API Endpoints:

Method	Endpoint	Description	Auth
GET	`/api/v1/DataRights/export`	Export user data (JSON)	User
GET	`/api/v1/DataRights/export/pdf`	Export user data (PDF)	User
POST	`/api/v1/DataRights/rectification`	Request data correction	User
POST	`/api/v1/DataRights/deletion`	Request account deletion	User
GET	`/api/v1/DataRights/requests`	View request history	User
POST	`/api/v1/DataRights/admin/deletion/{id}/process`	Process deletion	Admin

Data Exported:

- User profile (name, email, role, creation date)
- Organization memberships
- VVN summary (IDs and statuses)
- Staff assignments
- Privacy policy consent history

Frontend:

- `/profile/data-rights` - 4-tab interface:
    1. **Export Tab** - Download JSON/PDF
    2. **Rectification Tab** - Submit correction requests
    3. **Deletion Tab** - Request account deletion
    4. **History Tab** - View all requests and statuses

GDPR Articles Addressed:

- Article 15 (Right of access)
- Article 16 (Right to rectification)
- Article 17 (Right to erasure)
- Article 20 (Right to data portability)

C. Non-User Data Requests (US 4.5.4)

Purpose: Allow individuals who are not registered users (e.g., crew members, captains, representatives) to exercise their GDPR rights.

Domain Layer:

- Reuses `DataRequest` entity with `Source = EXTERNAL`
- `DataRequestType` enum includes: ACCESS, RECTIFICATION, ERASURE

API Endpoints:

Method	Endpoint	Description	Auth

Method	Endpoint	Description	Auth
POST	`/api/v1/DataRequests/non-user`	Submit non-user request	Public
GET	`/api/v1/DataRequests/non-user/status`	Check request status	Public
GET	`/api/v1/DataRequests`	List all requests	Admin
GET	`/api/v1/DataRequests/pending`	List pending requests	Admin
PUT	`/api/v1/DataRequests/{id}/status`	Update request status	Admin

Frontend:

- `/privacy-policy` - Includes non-user request form
- Form fields: Name, Email, CitizenId, Request Type, Description
- Status check functionality

GDPR Articles Addressed:

- Article 12 (Transparent communication)
- Article 15, 16, 17 (Data subject rights for non-users)

D. Implementation Architecture Summary

```
Frontend (Angular)
├── /privacy-policy          → Public policy display + Non-user form
├── /profile/data-rights     → User data rights (4 tabs)
└── Navbar dropdown          → "My Data Rights" link

Core Backend (.NET)
├── Domain/
│   ├── PrivacyPolicies/     → Policy + Acknowledgment entities
│   └── DataRequests/        → DataRequest entity (USER + EXTERNAL)
├── Application/
│   ├── Services/
│   │   ├── PrivacyPolicyService.cs
│   │   ├── DataRequestService.cs
│   │   └── DataRightsService.cs
│   └── DTOs/
│       ├── PrivacyPolicyDtos.cs
│       ├── DataRequestDtos.cs
│       └── DataRightsDtos.cs
└── Presentation/Controllers/
    ├── PrivacyPolicyController.cs
    ├── DataRequestsController.cs
    └── DataRightsController.cs

Database (SQL Server)
├── PrivacyPolicies table
├── PrivacyPolicyAcknowledgments table
└── DataRequests table (extended)
```

---

2. Sprint C Technical Implementation - GDPR Perspective

2.1. OEM Backend Architecture (US 4.1.1)

Technology Stack:

- **Runtime:** Node.js 20+
- **Language:** TypeScript 5+
- **Framework:** Express.js 4.x
- **Database:** MongoDB (NoSQL document database)
- **Architecture:** Clean Architecture with Domain-Driven Design (DDD)

GDPR-Relevant Design Decisions:

A. Clean Architecture Layers

```
Presentation (REST API) → Application (Services) → Domain (Entities) ←
Infrastructure (MongoDB, HTTP Clients)
```

Benefit for GDPR:

- Domain entities enforce business rules (e.g., audit logging on updates)
- Repository pattern abstracts data access (easier to implement encryption, pseudonymization)
- Infrastructure layer is isolated (database migration doesn't affect business logic)

Concern:

- No **Data Protection layer** - encryption, anonymization, retention are not centralized

B. No Direct Database Access Between Services

Rule: OEM Backend **never queries** Core Backend's SQL Server database directly. All data exchange happens via REST API calls.

GDPR Benefits:

- ☑ Clear data access boundaries (audit logging at API gateway)
- ☑ Access control enforced by Core Backend (RBAC/ABAC)
- ☑ Data minimization (OEM only receives data it explicitly requests)

GDPR Concerns:

- ⚠ API calls not logged in OEM (no record of when OEM accessed user data from Core)
- ⚠ Network interception risk (HTTPS mandatory, but no mutual TLS)
- ❌ No API rate limiting (potential for excessive data access)

C. JWT-Based Authentication

Mechanism:

1. Frontend authenticates with Core Backend (Google OAuth)
2. Core Backend issues JWT access token (60 min) and refresh token (7 days)
3. Frontend includes JWT in `Authorization: Bearer <token>` header for OEM API calls
4. OEM validates JWT signature using **shared JWT secret** with Core Backend
5. OEM extracts user claims (userId, email, name, role, orgId) from JWT
6. OEM enforces authorization rules based on role

GDPR Considerations:

- **Shared Secret:** Core Backend and OEM share the same `JWT_SECRET` environment variable
  - ☑ Enables stateless authentication
  - ❌ Secret compromise affects both systems
  - ❌ No key rotation policy
- **User Data in JWT:** Token contains `email` and `name` (personal data)
  - ⚠ Token transmitted with every API call (exposure risk)
  - ⚠ Token stored in browser localStorage (XSS vulnerability - inherited from Sprint B)
- **No Token Revocation:** If user is deactivated in Core Backend, their JWT remains valid until expiration
  - ❌ Cannot immediately revoke access for terminated staff

Inherited Risk from Sprint B: JWT token security gaps apply to OEM as well

---

2.2. Operation Plan Management (US 4.1.2, 4.1.3, 4.1.4)

Purpose: Store and manage planning results generated by scheduling algorithms (or manually created by operators)

Personal Data Collected:

- `createdBy` (User ID - who generated the plan)
- `operations[].assignedStaff[]` (Staff IDs - planned resource allocation)
- `auditLog[].userId` (User ID - who modified the plan)
- `auditLog[].userName` (User Name - **CACHED PERSONAL DATA**)

Data Flow:

1. Logistics Operator selects target date + scheduling algorithm
2. Frontend calls Planning Service (Prolog) or OEM directly
3. Planning result includes VVN IDs, staff assignments, crane allocations, time windows
4. Operator reviews plan in SPA
5. Operator clicks "Save" → OEM creates OperationPlan document in MongoDB
6. `createdBy` field populated from JWT token `userId` claim
7. Initial audit log entry created (action: CREATED, userId, userName, timestamp)

GDPR Compliance Assessment:

☑ **Data Minimization:**

- Only stores user IDs, not full user profiles

- Staff assignments are IDs only

⚠ **Purpose Limitation:**

- Plans are retained indefinitely (no retention policy)
- Old plans with staff assignments may no longer be necessary after 1 year

⚠ **Accuracy:**

- If staff member is terminated, their ID remains in historical plans
- No mechanism to update historical records if staff data changes

❌ **Audit Trail Violation:**

- `auditLog[].userName` duplicates personal data from Core Backend
- **Justification:** Performance optimization (avoid API call for every audit log display)
- **Problem:** If user changes name in Core Backend, audit logs in OEM show stale name
- **Article 5(1)(d) Violation:** Inaccurate personal data persisted

US 4.1.4 - Update Operation Plan with Audit Logging:

Implementation:

```
// domain/entities/OperationPlan.ts
private _auditLog: OperationPlanAuditEntry[];

public logChange(userId: string, userName: string, changes: Change[], reason?:
string): void {
  this._auditLog.push({
    timestamp: new Date(),
    userId,
    userName,  // GDPR CONCERN: Caching personal data
    action: 'UPDATED',
    changes,
    reason
  });
}
```

GDPR Analysis:

- ☑ **Accountability (Article 5(2)):** Change tracking implemented
- ☑ **Article 30 (Records of Processing):** Audit logs support compliance
- ❌ **Article 5(1)(c) (Data Minimization):** `userName` is unnecessary (could fetch on-demand)
- ❌ **Article 5(1)(e) (Storage Limitation):** Audit logs retained indefinitely

Recommendation:

- **Option 1:** Remove `userName` from audit log, fetch from Core Backend when displaying
- **Option 2:** Keep `userName` but document as "personal data snapshot at time of action" (requires privacy policy disclosure)
- **Option 3:** Implement audit log retention policy (e.g., 5 years per maritime regulations)

2.3. Vessel Visit Execution Management (US 4.1.7-4.1.11)

Purpose: Record actual execution of vessel visits (what really happened vs. what was planned)

Personal Data Collected:

- User ID of VVE creator
- User ID of operator who completes VVE
- Staff IDs of workers who executed cargo operations

Data Flow Example (US 4.1.9 - Update VVE with Executed Operations):

1. Logistics Operator observes cargo operation completion
2. Operator opens VVE in SPA
3. Operator updates operation: actual start time, end time, containers processed, staff assigned
4. Frontend sends PATCH request to OEM API with JWT token
5. OEM validates token, extracts `userId`
6. OEM updates `VesselVisitExecution.operations[]` array
7. Operation marked as "COMPLETED", staff IDs recorded

GDPR Considerations:

☑ **Legal Basis Strong:**

- Article 6(1)(c) - Legal Obligation (maritime safety requires execution tracking per IMO regulations)
- Article 6(1)(f) - Legitimate Interest (operational efficiency, performance analysis)

☑ **Data Minimization:**

- Only staff IDs stored, not full staff profiles

⚠ **Purpose Creep Risk:**

- Execution data used for performance analytics (e.g., "Staff A completes operations 10% faster than Staff B")
- **Original Purpose:** Safety and operational tracking
- **Secondary Purpose:** Performance evaluation (may require separate legal basis or consent)

⚠ **Employee Rights:**

- Staff members are **data subjects** whose work performance is being tracked
- Article 13: Staff must be informed their assignments are recorded in OEM
- Article 15: Staff have right to access their execution history
- **No SAR mechanism for staff** to request their data

❌ **No Data Retention Policy:**

- VVE records retained indefinitely
- Staff assignments from 5 years ago still accessible
- Maritime law may require 5-year retention, but beyond that, data should be anonymized

US 4.1.11 - Complete VVE (Immutability Concern):

Implementation:

- Once VVE status = "COMPLETED", record becomes read-only (except for admin corrections)
- Immutability ensures audit trail integrity

GDPR Conflict:

- **Right to Erasure (Article 17):** If staff member requests deletion, completed VVE cannot be modified
- **Right to Rectification (Article 16):** If staff member's ID changes, historical records cannot be updated
- **Resolution:** Legal grounds for refusal (Article 17(3)(b)) - compliance with legal obligation (maritime records)

2.4. Incident Management (US 4.1.12, 4.1.13)

Purpose: Track operational disruptions, safety events, and unexpected incidents affecting port operations

Personal Data Collected:

- `reportedBy` (User ID or name - incident reporter)
- `investigatedBy` (User ID - investigator)
- `resolvedBy` (User ID - resolver)
- `closedBy` (User ID - closer)
- `notes[].author` (User ID or name - note author)
- `externalEntitiesInvolved[]` (**CRITICAL: May contain personal data**)

Example Incident Record:

```
{
  "incidentId": "INC-2026-001",
  "incidentTypeId": "CRANE-MALFUNCTION",
  "title": "Crane 3 hydraulic failure",
  "description": "Crane 3 stopped mid-operation due to hydraulic pressure loss",
  "severity": "MAJOR",
  "status": "RESOLVED",
  "reportedAt": "2026-01-03T08:15:00Z",
  "reportedBy": "user-abc-123",
  "investigatedBy": "user-def-456",
  "resolvedBy": "user-def-456",
  "externalEntitiesInvolved": ["Harbor Master", "Equipment Vendor - TechCrane
Inc"],
  "affectedVves": ["VVE-2026-PT-001", "VVE-2026-PT-002"],
  "resolutionSummary": "Hydraulic pump replaced by TechCrane technician João
Silva"
}
```

GDPR Critical Issues:

❌ **Issue 1: Personal Data in Free-Text Fields**

- **Field:** `description`, `resolutionSummary`, `notes[].content`
- **Risk:** Operators may include personal data in free text (e.g., "Captain Smith refused to wait", "Staff member Silva was on break")
- **Article 5(1)(c) Violation:** Unstructured personal data collection violates data minimization
- **Recommendation:**
  - Add data entry guidelines ("Do not include individual names in incident descriptions")
  - Implement automated PII detection and redaction
  - Provide structured fields for involved parties

❌ Issue 2: External Entities as Third-Party Data

- **Field:** `externalEntitiesInvolved[]`
- **Example:** "Inspector João Silva - Customs"
- **Problem:** Individual names of external agents are personal data
- **Data Controller Question:** Who controls this data?
  - If Customs provided the inspector name → Customs is data controller
  - If Port recorded the name → Port is controller (requires legal basis)
- **Article 6 Violation:** No legal basis established for collecting external entity personal data
- **Article 13 Violation:** External entities not informed their names are recorded
- **Recommendation:**
  - Standardize to entity names only (no individuals): "Customs Authority", "Fire Department"
  - If individual tracking is necessary, obtain consent or establish legal obligation
  - Draft data sharing agreements with frequent external entities

❌ Issue 3: Incident Data Shared with Shipping Agents

- **Scenario:** Shipping agent requests incident report affecting their vessel
- **Data Sharing:** Report includes `reportedBy`, `investigatedBy`, `externalEntitiesInvolved`
- **Problem:** Sharing port staff user IDs and external entity names with third party
- **Article 6 Requirement:** Legitimate interest assessment required, or explicit consent
- **Article 13 Requirement:** Staff must be informed their involvement in incidents may be disclosed
- **Recommendation:** Anonymize incident reports for external sharing (remove user IDs, generalize entities)

⚠ Issue 4: Incident Retention for Litigation

- **Purpose:** Incidents may be evidence in legal disputes (insurance claims, liability cases)
- **Retention Period:** Legal requirements may extend beyond GDPR general retention limits
- **Article 17(3)(e):** Right to erasure doesn't apply if data needed for legal claims
- **Recommendation:** Document retention policy: "Incidents retained for 10 years for legal defense purposes"

2.5. Complementary Task Management (US 4.1.14, 4.1.15)

Purpose: Track non-cargo activities during vessel visits (inspections, maintenance, cleaning)

Personal Data Collected:

- `assignedStaff[]` (Staff IDs - workers assigned to task)

- `assignedTeam` (Team name - may include personal identifiers if team named after supervisor)
- `completedBy` (User ID - task completor)

GDPR Considerations:

☑ **Legal Basis:**

- Article 6(1)(b) - Contract (employment - staff assignments)
- Article 6(1)(c) - Legal Obligation (safety tasks must be documented)

⚠ **Team Name Concern:**

- **Field:** `assignedTeam` (string, free-text)
- **Example:** "Silva's Maintenance Team", "João's Cleaning Crew"
- **Problem:** Team names may contain personal data (supervisor names)
- **Recommendation:** Standardize team names to functional descriptions ("Maintenance Team A", "Cleaning Crew 1")

⚠ **Task Performance Metrics:**

- Tasks record `startTime` and `endTime`
- Can derive staff productivity metrics (e.g., "Staff A completes cleaning tasks 20% faster")
- **Purpose Creep:** If used for performance reviews, requires separate legal basis
- **Employee Rights:** Staff should be informed tasks are timed and may affect evaluations

---

3. GDPR Principles Assessment - Sprint C

GDPR Principle	Sprint A Status	Sprint B Status	Sprint C Status	Gap Assessment
Lawfulness, Fairness, Transparency	✗ No privacy policy	⚠ Google OAuth partial	☑ **Privacy Policy implemented (US 4.5.1)**	Privacy policy available at `/privacy-policy`, versioning supported
Purpose Limitation	⚠ Implicit purposes	⚠ OAuth purposes clear	☑ **Purposes documented in privacy policy**	Privacy policy details data collection purposes
Data Minimisation	☑ Necessary data	⚠ JWT contains excess data	⚠ `auditLog.userName` duplicates data	Remove cached personal data; FAL Convention Article 1.1 alignment
Accuracy	☑ Update methods exist	☑ Google profile verified	☑ **Rectification requests implemented (US 4.5.3)**	Users can request data corrections
Storage Limitation	✗ No retention policy	✗ No retention policy	⚠ No MongoDB retention policy	Critical gap worsens with dual databases

GDPR Principle	Sprint A Status	Sprint B Status	Sprint C Status	Gap Assessment
Integrity and Confidentiality	⚠ No encryption at rest	⚠ Token storage vulnerable	⚠ MongoDB not encrypted	MongoDB supports encryption (not enabled)
Accountability	⚠ Basic audit logging	⚠ Incomplete audit logs	☑ **Consent tracking implemented (US 4.5.2)**	User consent logged with timestamps

3.1. Data Minimization - FAL Convention Alignment

FAL Convention Article 1.1 (Norma): "As autoridades públicas não deverão exigir, seja qual for o caso, senão as informações indispensáveis e deverão reduzir o seu número a um mínimo."

Translation: "Public authorities shall not require, in any case, more than essential information and shall reduce their number to a minimum."

Analysis:

- ☑ The FAL Convention data minimization principle **aligns perfectly** with GDPR Article 5(1)(c) Data Minimization
- ☑ **Compliance:** VVN data collection (captain/crew names, CitizenId) is justified by FAL Convention Article 2.6 (crew list requirements) and Article 2.7 (passenger list requirements)
- ☑ **OEM Backend Design:** Storing only user IDs and staff IDs (not full profiles) follows both FAL Convention and GDPR minimization principles
- ⚠ **Gap Identified:** The OEM module collects `auditLog[].userName` beyond FAL requirements - **violates both GDPR and FAL Convention**
- ⚠ **External Entity Names:** FAL Convention Article 4 requires cooperation with external entities but does not mandate storing individual inspector names - entity name only is sufficient

Recommendation:

1. Audit all personal data fields against FAL Convention requirements
2. Remove `auditLog[].userName` field (already recommended in Section 6.1 Action 2)
3. Standardize external entity references to official entity names only
4. Document FAL Convention justification for each VVN personal data field in privacy policy

---

4. Data Storage and Security Analysis

4.1. MongoDB as Personal Data Store

MongoDB Collections Containing User References:

1. `operationplans` - User IDs in `createdBy`, `auditLog[]`
2. `vesselvisitexecutions` - User IDs in `completedBy`, staff IDs in operations
3. `incidents` - User IDs in reporter/investigator/resolver fields, external entity names
4. `complementarytasks` - User IDs in `completedBy`, staff IDs in assignments

Security Assessment:

⚠ **MongoDB Encryption at Rest:**

- **Capability:** MongoDB supports encryption at rest (Enterprise Edition or MongoDB Atlas)
- **Current Status:** Not enabled (per .env.example, local MongoDB instance)
- **GDPR Article 32:** "Encryption of personal data" required for appropriate security
- **Risk:** Database file theft exposes all user IDs, incident data, external entity names
- **Recommendation:** Enable MongoDB encryption at rest immediately (priority: HIGH)

⚠ **MongoDB Authentication:**

- **Current Configuration:** `DATABASE_URL=mongodb://localhost:27017/oem` (no authentication)
- **Risk:** Anyone with network access can read/write OEM database
- **GDPR Article 32:** "Means of ensuring ongoing confidentiality" required
- **Recommendation:**
  - Enable MongoDB authentication (SCRAM-SHA-256)
  - Use strong passwords
  - Restrict network access (bind to localhost only or use firewall)

⚠ **Field-Level Encryption:**

- **Capability:** MongoDB supports Client-Side Field Level Encryption (CSFLE)
- **Use Case:** Encrypt sensitive fields like `externalEntitiesInvolved` at application level
- **Current Status:** Not implemented
- **Benefit:** Even if database is compromised, encrypted fields remain protected
- **Recommendation:** Implement CSFLE for high-risk fields in Phase 2

☑ **MongoDB Audit Logging:**

- **Capability:** MongoDB Enterprise supports audit logging (all database operations)
- **Current Status:** Not enabled (Community Edition in use)
- **Benefit:** Track all data access, modifications, deletions for GDPR Article 30 compliance
- **Recommendation:** Consider MongoDB Enterprise or implement application-level audit logging

---

4.2. Dual Database Architecture (SQL Server + MongoDB)

Personal Data Distribution:

- **SQL Server (Core Backend):** User profiles, staff details, crew CitizenId, representative CitizenId
- **MongoDB (OEM Backend):** User IDs, staff IDs, audit trails, incident notes

GDPR Challenges:

❌ **Challenge 1: Coordinated Data Erasure**

- **Scenario:** User exercises Right to Erasure (Article 17)
- **Required Action:**
  1. Delete user from SQL Server (Core Backend)
  2. Delete/anonymize user IDs in MongoDB (OEM Backend)

- **Problem:** No automated cross-database deletion mechanism
- **Risk:** Partial deletion leaves orphaned data in MongoDB
- **Recommendation:** Implement coordinated deletion API endpoint

❌ **Challenge 2: Distributed Audit Trail**

- **Example:** User updates operation plan → OEM logs in MongoDB
- **Example:** User accesses VVN details → Core logs in SQL Server
- **Problem:** Complete user activity history requires querying both databases
- **Article 30 Requirement:** Controller must maintain records of all processing activities
- **Recommendation:** Centralized logging service or log aggregation (e.g., ELK stack, Splunk)

❌ **Challenge 3: Inconsistent Retention Policies**

- **Scenario:** Core Backend purges inactive users after 2 years
- **Problem:** User IDs in MongoDB remain active indefinitely
- **Result:** MongoDB contains references to non-existent users
- **Recommendation:** Synchronized retention policies across both systems

☑ **Challenge 4: Data Subject Access Requests (SARs) - IMPLEMENTED**

- **Article 15:** User requests all data held about them
- **Implementation (US 4.5.3):**
  - `GET /api/v1/DataRights/export` - JSON export of user data
  - `GET /api/v1/DataRights/export/pdf` - PDF export of user data
  - `POST /api/v1/DataRights/rectification` - Request data corrections
  - `POST /api/v1/DataRights/deletion` - Request account deletion
  - `GET /api/v1/DataRights/requests` - View request history
- **Current Scope:** Core Backend data (user profile, organizations, VVNs, staff assignments)
- **Future Enhancement:** Cross-service aggregation to include OEM Backend data (operation plans, incidents)

4.3. Inter-Service Communication Security

Data Flow: Frontend → OEM Backend → Core Backend (for VVN data, user details)

Current Implementation:

- HTTPS for all API calls (configured via `CORE_BACKEND_URL` environment variable)
- JWT token validation (shared secret between services)
- No mutual TLS (mTLS)
- No API gateway/service mesh

GDPR Considerations:

⚠ **Network Transit Security:**

- ☑ HTTPS encrypts data in transit
- ❌ No certificate pinning (man-in-the-middle attacks possible)
- ❌ No request/response logging for audit trails

- ❌ No rate limiting (excessive data access not prevented)

⚠ **Service Authentication:**

- Currently: OEM validates JWT from frontend, trusts Core Backend API responses implicitly
- **Gap:** Core Backend doesn't verify OEM's identity when OEM calls Core APIs
- **Risk:** If OEM is compromised, attacker can fetch unlimited user data from Core
- **Recommendation:** Implement service-to-service authentication (API keys, mTLS, or OAuth 2.0 client credentials)

---

5. Future Problems and Risks with Current Sprint C Implementation

5.1. Critical Risks

Risk 1: MongoDB Data Breach Exposing User Identifiers and Incident Data

Probability: High (no encryption at rest, no authentication enabled)

Impact: Severe

GDPR Fine Potential: €20 million or 4% of annual global turnover (Article 83(5))

Scenario:

1. Attacker gains access to MongoDB port 27017 (default, open)
2. No authentication required → attacker can connect and dump entire database
3. Exposed data:
    - 10,000+ user IDs (can be linked to Core Backend via API calls)
    - Incident reports with external entity names (potentially personal data)
    - Staff assignment records (employee IDs linked to work schedules)
    - Audit logs with user names (direct personal data)

Cascading Impact:

- User IDs used to query Core Backend API for full profiles (if API not rate-limited)
- External entity names leaked to public (reputational damage to customs inspectors, coast guard officers)
- Staff performance data exposed (labor law violations)

Mitigation Required (Immediate - 0-2 weeks):

1. Enable MongoDB authentication (SCRAM-SHA-256) with strong password
2. Configure MongoDB to listen only on localhost (127.0.0.1)
3. Enable MongoDB encryption at rest (requires MongoDB Enterprise or MongoDB Atlas with encryption)
4. Implement network firewall rules (allow only Core Backend and OEM Backend IPs)
5. Enable MongoDB audit logging (Enterprise feature or application-level logging)

Cost-Benefit:

- **Implementation Effort:** 2-3 developer days
- **Avoided Fine:** Potentially €20 million

- **ROI:** Extremely high

Risk 2: Personal Data in Incident Free-Text Fields (Unstructured PII)

Probability: Very High (operators not trained to avoid personal data in descriptions)

Impact: Moderate to Severe (depending on sensitivity of data included)

GDPR Issues: Articles 5(1)(c) Data Minimization, 32 Security of Processing

Scenario:

1. Operator creates incident: "Crane failure caused by operator José Silva's mistake"
2. Incident shared with shipping agent → third-party receives staff member's name
3. Staff member José Silva not informed his name was disclosed
4. José Silva requests data erasure → incident description cannot be edited (part of legal record)

Additional Examples:

- "Captain Smith refused to wait for berth clearance" (captain's personal behavior recorded)
- "Customs Inspector Maria Santos delayed inspection due to personal phone call" (external entity personal data)
- "Staff member on medication caused equipment mishandling" (health data - Article 9 special category)

GDPR Violations:

- **Article 5(1)(c):** Unnecessary personal data collected
- **Article 9:** Special category data (health) collected without consent or legal basis
- **Article 13:** Data subjects not informed their data was collected
- **Article 17:** Cannot fulfill erasure requests for data embedded in legal records

Mitigation Required (Short-term - 1-3 months):

1. **Immediate:** Training program for operators on GDPR-compliant incident reporting
2. **Technical Control:** Implement automated PII detection in free-text fields (NLP-based)
    - Flag names, email addresses, phone numbers, health terms
    - Warn operator before submission: "Possible personal data detected"
3. **Structured Fields:** Replace free-text with structured options where possible
    - "Incident caused by: Equipment | Weather | Human Error | Other"
    - If "Human Error," provide role-based options (no names): "Operator | Captain | Inspector"
4. **Data Sanitization:** Regular audits of incident database to redact personal data from historical records
5. **Privacy Notice:** Add consent checkbox: "I confirm no personal data is included in this incident description"

Risk 3: No Coordinated Data Retention Policy Across Core + OEM

Probability: Certain (policy not implemented)

Impact: High (GDPR Article 5(1)(e) violation)

GDPR Fine Potential: €10-20 million or 2-4% of annual turnover

Scenario:

1. Staff member João terminated in 2024
2. Core Backend retains João's user account for 2 years (inactive user policy)
3. OEM retains João's user ID in operation plans, incidents, tasks indefinitely
4. In 2028, Core Backend deletes João's account (ID: user-abc-123)
5. OEM database still contains user-abc-123 references in 500+ documents
6. New user registered, assigned same ID (ID reuse bug) → data linkage to wrong person
7. João exercises Right to Erasure → port authority cannot prove full deletion (OEM records still exist)

Compounding Issue:

- MongoDB aggregation queries fail (user-abc-123 no longer exists in Core Backend)
- Audit trails become unverifiable (cannot determine who made changes)
- Legal defense compromised (incident reports reference non-existent users)

Mitigation Required (Critical - 2-4 months):

Data Retention Policy for OEM Module:

Data Type	Retention Period	Justification	Disposal Method
Operation Plans	3 years	Operational efficiency analysis, dispute resolution	Anonymize user IDs, keep aggregated statistics
Vessel Visit Executions	7 years	Maritime legal requirements (IMO, insurance claims)	Archive to cold storage after 3 years, anonymize after 7 years
Incidents	10 years	Legal disputes, safety audits, regulatory compliance	Archive after 3 years, anonymize non-essential data after 10 years
Complementary Tasks	2 years	Operational efficiency only	Hard delete after 2 years
Audit Logs (Operation Plans)	5 years	Internal audit, GDPR accountability	Anonymize user names, keep action types only
Audit Logs (Incidents)	10 years	Legal defense, regulatory compliance	Archive after 5 years

Implementation Approach:

1. **Scheduled Job:** MongoDB cron job or Node.js scheduled task (node-cron)
2. **Anonymization Function:**

```
async anonymizeUser(userId: string): Promise<void> {
  await db.collection('operationplans').updateMany(
    { createdBy: userId },
```

```
      { $set: { createdBy: 'ANONYMIZED', 'auditLog.$[].userName': 'ANONYMIZED'
  } }
    );
    await db.collection('incidents').updateMany(
      { reportedBy: userId },
      { $set: { reportedBy: 'ANONYMIZED' } }
    );
    // ... repeat for all collections
  }
```

3. **Coordination with Core Backend:** When Core deletes user, trigger OEM anonymization via API call
4. **Legal Review:** Confirm retention periods comply with Portuguese maritime law, labor law, and insurance requirements

---

Risk 4: Third-Party Data Sharing via Incident Reports

Probability: High (shipping agents request incident reports)

Impact: Moderate to High

GDPR Issue: Article 6 (no legal basis for disclosure), Article 13 (no transparency)

Scenario:

1. Incident recorded: "Fire Department responded to fuel leak, Inspector Carlos handled containment"
2. Shipping agent requests incident report affecting their vessel
3. Port authority shares full incident report via email
4. Inspector Carlos (Fire Department employee) not informed his name was shared with private company
5. Inspector Carlos's employer (Fire Department) not consulted about data sharing

Legal Analysis:

- **Who is the data controller?**
  - Port authority collected Inspector Carlos's name
  - Fire Department may argue they control their employee data
  - **Joint Controllers (Article 26):** Port and Fire Department share control → requires formal agreement
- **Legal basis for sharing with shipping agent:**
  - Article 6(1)(f) Legitimate Interest? → Requires balancing test
  - Article 6(1)(b) Contract? → Only if incident report is contractually required
  - **No valid basis if not assessed**

Mitigation Required (Short-term - 1-2 months):

1. **Privacy Notice for External Entities:**
   - When external entity responds to incident, provide notice: "Your involvement will be recorded and may be shared with affected parties per legal obligations"
2. **Anonymized Incident Reports for Third Parties:**
   - Public version: "Fire Department responded" (no individual names)

    - Full version: Retained internally for legal defense only
3. **Data Sharing Agreements:**
    - Establish framework agreements with frequent external entities (Customs, Fire Department, Coast Guard)
    - Define data sharing scope, legal basis, retention periods
4. **Incident Report Disclosure Policy:**
    - Document when and how incident reports may be shared
    - Implement approval workflow (supervisor must approve disclosure)
    - Log all disclosures for GDPR Article 30 compliance

5.2. High Risks

Risk 5: JWT Token Compromise Affecting Both Systems

Probability: Moderate (inherited from Sprint B)

Impact: High

Sprint C Amplification: OEM Backend shares the same JWT secret as Core Backend. A compromised token grants access to BOTH systems.

Attack Scenario:

1. Attacker exploits XSS vulnerability in frontend (localStorage access)
2. Steals JWT access token containing `userId`, `email`, `name`, `role: "Admin"`
3. Token valid for 60 minutes
4. Attacker uses token to:
    - Access all user data via Core Backend APIs
    - Access all operation plans, incidents, VVEs via OEM Backend APIs
    - Create/modify/delete data in both systems as admin

Mitigation (inherited from Sprint B, applies to Sprint C):

- Move refresh tokens to httpOnly cookies (prevents XSS access)
- Implement Content Security Policy (CSP) headers (mitigates XSS)
- Implement token revocation mechanism (Redis blacklist)
- Reduce token lifetime (30 minutes instead of 60)
- Implement anomaly detection (multiple IPs using same token)

Risk 6: No Service-Level Data Processing Agreement (DPA)

Probability: Certain (not documented)

Impact: Moderate (regulatory compliance gap)

GDPR Issue: Article 28 or Article 26 (depending on controller/processor relationship)

Scenario Analysis:

Option A: OEM is a Processor

- If Core Backend determines purposes and means of processing
- OEM simply executes instructions from Core Backend
- **Requirement:** Article 28 Data Processing Agreement (DPA)
- **DPA Must Include:**
  - Processing scope and duration
  - Nature and purpose of processing
  - Types of personal data
  - Categories of data subjects
  - Processor obligations (security, sub-processors, audits)
  - Data breach notification (processor must notify controller within 24 hours)

Option B: Core and OEM are Joint Controllers

- If both determine purposes and means
- Example: OEM decides to retain operation plans for 3 years (not instructed by Core)
- **Requirement:** Article 26 Joint Controller Agreement
- **Agreement Must Include:**
  - Respective responsibilities for GDPR compliance
  - Who handles data subject rights requests (access, erasure, rectification)
  - Who provides privacy notices
  - Contact points for supervisory authority

Current Reality:

- OEM makes independent decisions (e.g., what data to log in audit trails, retention periods)
- Core Backend doesn't "instruct" OEM on how to process data
- **Conclusion:** Likely **Joint Controllers** scenario

Mitigation Required (Medium-term - 1-2 months):

1. Conduct legal analysis of controller/processor roles
2. Draft Joint Controller Agreement or DPA (depending on analysis outcome)
3. Define responsibilities:
   - Core Backend: User authentication, profile management, SAR fulfillment for personal data
   - OEM Backend: Operation tracking, incident management, SAR fulfillment for execution data
4. Establish coordination procedures for data subject rights
5. Document agreement and make available to supervisory authority (CNPD)

Risk 7: Performance Metrics Derived from Staff Execution Data

Probability: High (data enables analytics)

Impact: Moderate (employee rights, labor law)

GDPR Issue: Purpose creep (Article 5(1)(b)), employee consent (Article 6)

Scenario:

1. VVE records show:
    - Staff A completes 150 container operations per shift (average time: 1.2 minutes/container)
    - Staff B completes 120 container operations per shift (average time: 1.5 minutes/container)
2. Supervisor uses OEM data to generate performance report
3. Staff B receives negative performance review based on OEM data
4. Staff B was not informed their work would be monitored for performance evaluation

Legal Analysis:

- **Original Purpose:** Safety and operational tracking (legal obligation - Article 6(1)(c))
- **Secondary Purpose:** Performance evaluation (requires separate legal basis)
- **Options:**
    - **Consent:** Not valid for employment (power imbalance per EDPB guidelines)
    - **Legitimate Interest:** Possible, but requires balancing test and employee notification
    - **Contract:** If performance monitoring is in employment contract

Employee Rights (GDPR + Labor Law):

- **Article 13:** Right to be informed about monitoring
- **Article 15:** Right to access their performance data
- **Article 22:** Right to human review if automated decisions affect them
- **Portuguese Labor Code:** Specific rules on employee monitoring

Mitigation Required (Medium-term - 2-3 months):

1. **Transparency:**
    - Update employee handbook: "Work assignments and completion times are recorded in OEM system for operational planning and may be used for performance evaluation"
    - Provide staff with access to their own execution history (self-service portal)
2. **Purpose Documentation:**
    - Formally document dual purposes: (1) operational tracking, (2) performance evaluation
    - Conduct legitimate interest assessment (DPIA) for performance monitoring
3. **Anonymized Analytics:**
    - For operational efficiency analysis, use anonymized/aggregated data only
    - Example: "Average operation time: 1.35 min/container" (no individual breakdown)
4. **Data Minimization:**
    - Don't retain individual-level execution data longer than necessary for performance evaluation period (e.g., 1 year)

---

6. Improvements and Recommendations

6.1. Immediate Actions (Priority: CRITICAL - 0-4 weeks)

Action 1: Secure MongoDB Database

Timeline: 1 week
Effort: 1-2 developer days
GDPR Articles: 32 (Security of Processing)

Implementation Steps:

1. Enable MongoDB authentication:

```
# mongo shell
use admin
db.createUser({
  user: "oemAdmin",
  pwd: "STRONG_PASSWORD_HERE",
  roles: [ { role: "readWrite", db: "oem" } ]
})
```

2. Update `DATABASE_URL` in .env:

```
DATABASE_URL=mongodb://oemAdmin:PASSWORD@localhost:27017/oem?
authSource=admin
```

3. Configure MongoDB to bind to localhost only:

```
# mongod.conf
net:
  bindIp: 127.0.0.1
```

4. Enable firewall rules (if cloud-hosted)

Action 2: Remove Cached Personal Data from Audit Logs

Timeline: 2 weeks
Effort: 2-3 developer days
GDPR Articles: 5(1)(c) Data Minimization

Implementation:

1. Remove `userName` field from `OperationPlanAuditEntry`:

```
export interface OperationPlanAuditEntry {
  timestamp: Date;
  userId: string;  // Keep
  // userName: string;  // REMOVE
  action: string;
  changes: Change[];
  reason?: string;
}
```

2. Update frontend to fetch user names on-demand:

```
// When displaying audit log:
const userName = await coreBackendApi.getUserName(auditEntry.userId);
```

3. Migration script to remove existing `userName` values:

```
await db.collection('operationplans').updateMany(
  {},
  { $unset: { 'auditLog.$[].userName': '' } }
);
```

Trade-off: Slight performance overhead (additional API calls), but GDPR compliant

---

Action 3: Implement Operator Training on GDPR-Compliant Incident Reporting

Timeline: 2 weeks
Effort: 1 day (create training materials + 1-hour training session)
GDPR Articles: 5(1)(c) Data Minimization, 13 Transparency

Training Content:

- ❌ **Bad:** "Crane operator João Silva made a mistake"
- ☑ **Good:** "Crane operator error (user ID: STAFF-123)"
- ❌ **Bad:** "Captain refused to follow instructions due to personal issues"
- ☑ **Good:** "Captain non-compliance with port instructions"
- ❌ **Bad:** "Customs Inspector Maria Santos delayed inspection"
- ☑ **Good:** "Customs inspection delay"

Enforcement:

- Add warning banner in incident creation form: "Do not include personal names or sensitive information"
- Implement review process: Supervisor approves incident reports before finalization

---

6.2. Short-Term Actions (Priority: HIGH - 1-3 months)

Action 4: Enable MongoDB Encryption at Rest

Timeline: 1-2 months (requires MongoDB Enterprise or cloud migration)
Effort: 3-5 developer days
GDPR Articles: 32 (Security of Processing)

Options:

Option A: MongoDB Enterprise Edition

- Purchase MongoDB Enterprise license
- Enable encryption at rest with KMIP key management
- Cost: $$$

Option B: MongoDB Atlas (Cloud)

- Migrate to MongoDB Atlas cloud service
- Encryption at rest included
- Automatic backups, monitoring, scaling
- Cost: $$-$$$ (usage-based)

Option C: Operating System-Level Encryption

- Linux: LUKS (dm-crypt)
- Windows: BitLocker
- Encrypts entire disk/volume
- Cost: Free
- **Limitation:** Less granular than MongoDB-native encryption

Recommendation: Option B (MongoDB Atlas) for production, Option C for development/staging

---

Action 5: Implement Coordinated Data Retention and Deletion

Timeline: 2-3 months
Effort: 5-7 developer days
GDPR Articles: 5(1)(e) Storage Limitation, 17 Right to Erasure

Architecture:

```
Core Backend (User Deletion) → Event Bus → OEM Backend (Anonymization)
```

Implementation Steps:

1. **Core Backend - Add Deletion Hook:**

```
// Application/Services/UserService.cs
public async Task DeleteUser(Guid userId)
{
    await _userRepository.Delete(userId);
    await _messageBus.Publish(new UserDeletedEvent { UserId = userId });
}
```

2. **OEM Backend - Subscribe to Deletion Events:**

```
// application/subscribers/UserDeletionSubscriber.ts
messageBus.subscribe('UserDeleted', async (event) => {
```

```
  await anonymizeUserData(event.userId);
});
```

3. **OEM Backend - Anonymization Service:**

```
async function anonymizeUserData(userId: string): Promise<void> {
  await operationPlanRepo.anonymizeUser(userId);
  await vveRepo.anonymizeUser(userId);
  await incidentRepo.anonymizeUser(userId);
  await taskRepo.anonymizeUser(userId);

  // Log deletion for audit
  await auditLog.create({
    action: 'USER_DATA_ANONYMIZED',
    userId: 'SYSTEM',
    details: `Anonymized all data for user ${userId}`,
    timestamp: new Date()
  });
}
```

4. **Scheduled Retention Job:**

```
// Runs nightly at 2 AM
cron.schedule('0 2 * * *', async () => {
  const threeYearsAgo = new Date();
  threeYearsAgo.setFullYear(threeYearsAgo.getFullYear() - 3);

  // Anonymize old operation plans
  await operationPlanRepo.anonymizeOlderThan(threeYearsAgo);

  // Delete old complementary tasks
  await taskRepo.deleteOlderThan(twoYearsAgo);
});
```

Alternative (Simpler): REST API endpoint in OEM

```
// POST /api/v1/users/:userId/anonymize
// Called by Core Backend when user is deleted
```

Action 6: Implement Automated PII Detection in Free-Text Fields

Timeline: 2-3 months
Effort: 5-7 developer days
GDPR Articles: 5(1)(c) Data Minimization, 25 Data Protection by Design

Technical Approach:

Option A: Regex-Based Detection (Simple)

```
function detectPII(text: string): string[] {
  const patterns = {
    email: /\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b/g,
    phone: /\b\d{9,11}\b/g,  // Portuguese phone numbers
    name: /\b[A-Z][a-z]+ [A-Z][a-z]+\b/g,  // Capitalized names
  };

  const detected = [];
  if (patterns.email.test(text)) detected.push('Email address');
  if (patterns.phone.test(text)) detected.push('Phone number');
  if (patterns.name.test(text)) detected.push('Possible personal name');

  return detected;
}
```

Option B: NLP-Based Detection (Advanced)

- Use spaCy (Python) or compromise (JavaScript) for Named Entity Recognition (NER)
- Detect: PERSON, EMAIL, PHONE, LOCATION
- More accurate but requires model training

Frontend Integration:

```
// When operator types in incident description:
onIncidentDescriptionChange(text) {
  const piiDetected = detectPII(text);
  if (piiDetected.length > 0) {
    showWarning(`Possible personal data detected: ${piiDetected.join(', ')}.
                 Please remove names and contact information.`);
  }
}
```

Action 7: Align VVN Data Collection with FAL Convention Standards

Timeline: 2-3 months **Effort:** 3-5 developer days **GDPR Articles:** 5(1)(c) Data Minimization, 6(1)(c) Legal Obligation **Legal Basis:** FAL Convention (Decreto n.º 13/1990)

FAL Convention Standard Forms: The convention specifies standard forms for crew lists, passenger lists, and cargo declarations (FAL Convention Section 2.B, Articles 2.6-2.7).

Implementation:

1. **Review Current VVN Personal Data Fields:**

- Compare each VVN field with FAL Convention Article 2.6.1 (crew list requirements) and Article 2.7.3 (passenger list requirements)
- Remove any fields not required by FAL Convention or other legal obligations
- Example: If collecting crew member's home address, this exceeds FAL requirements → remove unless justified by another law

2. **FAL Convention Mandatory Crew Data (Article 2.6.1):**

   - Name (family name + given names)
   - Nationality
   - Rank/Position
   - Date and place of birth
   - Nature and number of identity document
   - Port and date of arrival
   - **Not Required:** Email, phone, home address, emergency contact (unless required by other regulations)

3. **FAL Convention Mandatory Passenger Data (Article 2.7.3):**

   - Name (family name + given names)
   - Nationality
   - Date of birth
   - Place of birth
   - Port of embarkation/disembarkation
   - **Not Required:** Email, phone, passport photo, travel history

4. **Standardize External Entity References:**

   - FAL Convention mentions: "Customs", "Health Authority", "Immigration", "Port Authority"
   - Create controlled vocabulary based on FAL Convention entities:
     - "Portuguese Customs Authority" (not "Inspector João Silva")
     - "Fire Department" (not "Firefighter Team Leader Maria Santos")
     - "Coast Guard" (not "Captain Carlos Costa")
   - Prevent free-text entry of entity names (use dropdown with predefined options)

5. **Document Legal Justification:**

   - For each VVN personal data field, document: "Required by FAL Convention Article X.X" or "Required by [other law]"
   - If no legal requirement found → remove field (data minimization)
   - Create data mapping document showing FAL Convention compliance

Expected Outcome:

- Reduced data collection (GDPR Article 5(1)(c) compliance)
- Stronger legal defensibility (FAL Convention mandates provide Article 6(1)(c) legal basis)
- International standardization (FAL forms accepted by 116+ countries globally)
- Simplified data sharing with external entities (standardized entity names)

Reference: FAL Convention Articles 1.1, 2.6.1, 2.7.3, Decreto n.º 13/1990

Action 8: Establish Service-Level Agreement (Joint Controller or DPA)

Timeline: 1-2 months
Effort: Legal consultation (5-10 hours)
GDPR Articles: 26 (Joint Controllers) or 28 (Processor)

Recommended Approach: Joint Controller Agreement (Article 26)

Agreement Must Specify:

1. **Respective Responsibilities:**
    - Core Backend: User management, authentication, personal data master records
    - OEM Backend: Execution tracking, incident management, operational analytics
2. **Data Subject Rights Fulfillment:**
    - **Right to Access (Article 15):** Core Backend aggregates data from OEM and responds
    - **Right to Rectification (Article 16):** Core Backend updates user profile, OEM updates references if necessary
    - **Right to Erasure (Article 17):** Core Backend deletes user, triggers OEM anonymization
    - **Right to Data Portability (Article 20):** Core Backend exports personal data, OEM exports execution history
3. **Security Obligations:**
    - Both systems must maintain encryption at rest
    - Both systems must implement audit logging
    - Breach notification: Either party must notify the other within 12 hours
4. **Data Retention Coordination:**
    - Synchronized retention policies (see Section 6.2 Action 5)
5. **Contact Point for Data Subjects:**
    - Primary contact: Core Backend team (dpo@portauthority.pt)
    - Data subjects should not be aware of OEM/Core separation (transparency)

Template: Adapt EDPB's Joint Controller Agreement template

6.3. Medium-Term Actions (Priority: MEDIUM - 3-6 months)

Action 9: ☑ IMPLEMENTED - Subject Access Request (SAR) API

Status: Implemented in Sprint C (US 4.5.3) **GDPR Articles:** 15 (Right of Access), 16 (Right to Rectification), 17 (Right to Erasure), 20 (Right to Data Portability)

Implementation Details:

Backend (Core Backend .NET):

- `DataRightsService.cs` - Service layer for SAR operations
- `DataRightsController.cs` - REST API endpoints
- `DataRequest.cs` - Entity for tracking requests

API Endpoints:

```
GET  /api/v1/DataRights/export       - Export user data as JSON
GET  /api/v1/DataRights/export/pdf   - Export user data as PDF
POST /api/v1/DataRights/rectification - Submit rectification request
POST /api/v1/DataRights/deletion      - Submit deletion request
GET  /api/v1/DataRights/requests      - View user's request history
POST /api/v1/DataRights/admin/deletion/{id}/process - Admin processes deletion
```

Data Exported:

- User profile (name, email, role)
- Organization memberships
- VVN summary (created/associated VVNs)
- Staff assignments
- Consent history

Frontend:

- `data-rights.component.ts` - 4-tab UI (Export, Rectification, Deletion, History)
- Route: `/profile/data-rights`
- Navbar dropdown with "My Data Rights" link

Future Enhancement:

- Cross-service aggregation to include OEM Backend data (operation plans, incidents, tasks)
- Email notifications when request status changes

Action 10: Implement Token Revocation Mechanism

Timeline: 3-4 months
Effort: 3-5 developer days
GDPR Articles: 32 (Security of Processing)

Inherited from Sprint B, applies to Sprint C

Architecture:

```
Redis (Token Blacklist) ← Both Core and OEM backends check before accepting token
```

Implementation:

1. **Core Backend - Revoke Token on User Deactivation:**

```
public async Task DeactivateUser(Guid userId)
{
    user.Deactivate();
    await _redis.BlacklistAllTokensForUser(userId);
}
```

2. **OEM Backend - Check Blacklist:**

```
// middleware/jwtValidator.ts
async function validateToken(req, res, next) {
  const token = extractToken(req);
  const decoded = jwt.verify(token, JWT_SECRET);

  // Check if token is revoked
  const isRevoked = await redis.get(`revoked:${token}`);
  if (isRevoked) {
    return res.status(401).json({ error: 'Token revoked' });
  }

  req.user = decoded;
  next();
}
```

Action 11: Conduct Data Protection Impact Assessment (DPIA) for Incident Management

Timeline: 4-6 months
Effort: 10-15 hours (legal + technical)
GDPR Articles: 35 (Data Protection Impact Assessment)

DPIA Required Because:

- Incident management involves **external entities** (third-party personal data)
- Potential for **sensitive data** in free-text fields (health, criminal activity)
- **Sharing with shipping agents** (disclosure to third parties)

DPIA Process:

1. **Describe Processing:**
    - What: Recording incidents with external entity involvement
    - Why: Safety compliance, operational improvement, legal defense
    - How: MongoDB storage, API access, potential sharing with shipping agents
2. **Assess Necessity and Proportionality:**
    - Is external entity name necessary? → Yes (accountability, follow-up communication)
    - Is sharing with shipping agents necessary? → Depends on contractual obligations

3. **Identify Risks:**
    - Free-text fields may include sensitive data
    - External entities not informed of data collection
    - No consent or legal basis for third-party disclosure
4. **Mitigation Measures:**
    - Structured incident fields (minimize free-text)
    - PII detection and warning system
    - Anonymized incident reports for external sharing
    - Data sharing agreements with external entities
5. **Consult CNPD (if necessary):**
    - If residual high risk remains after mitigation, consult supervisory authority

Outcome: Documented DPIA report, approved by Data Protection Officer

---

6.4. Long-Term Actions (Priority: LOW - 6-12 months)

Action 12: Implement Field-Level Encryption for Incident External Entities

Timeline: 6-9 months
Effort: 7-10 developer days
GDPR Articles: 32 (Security of Processing)

MongoDB Client-Side Field Level Encryption (CSFLE):

```
// Encrypt externalEntitiesInvolved field
const clientEncryption = new ClientEncryption(client, {
  keyVaultNamespace: 'encryption.__keyVault',
  kmsProviders: { local: { key: masterKey } }
});

const dataKey = await clientEncryption.createDataKey('local');

const encryptedSchema = {
  'incidents': {
    'properties': {
      'externalEntitiesInvolved': {
        'encrypt': {
          'keyId': [dataKey],
          'algorithm': 'AEAD_AES_256_CBC_HMAC_SHA_512-Deterministic'
        }
      }
    }
  }
};
```

Benefit: Even if database is breached, external entity names remain encrypted

---

Action 13: Implement Centralized Audit Logging (ELK Stack or Splunk)

Timeline: 9-12 months
Effort: 15-20 developer days
GDPR Articles: 30 (Records of Processing Activities)

Architecture:

```
Core Backend → Elasticsearch ← OEM Backend
                    ↓
                Kibana (Visualization)
```

Benefits:

- Unified view of all data access across both systems
- Compliance reports (who accessed what, when)
- Anomaly detection (unusual access patterns)
- Article 30 compliance (records of processing activities)

---

7. Data Retention Policy Recommendation

7.1. Proposed Retention Periods for OEM Module

Data Category	Retention Period	Legal Basis	Anonymization After	Hard Delete After
Operation Plans	3 years	Operational analysis, dispute resolution	3 years (anonymize user IDs)	7 years (delete anonymized data)
Vessel Visit Executions	7 years	Maritime law (IMO), insurance claims, safety audits	5 years (anonymize staff IDs)	10 years (delete anonymized data)
Incidents - Safety/Environmental	10 years	Legal disputes, regulatory compliance, safety audits	7 years (anonymize reporters)	15 years (legal defense period)
Incidents - Minor Operational	3 years	Operational improvement only	3 years (hard delete)	N/A
Complementary Tasks	2 years	Operational efficiency	2 years (hard delete)	N/A

Data Category	Retention Period	Legal Basis	Anonymization After	Hard Delete After
Audit Logs (Operation Plans)	5 years	Internal audit, GDPR accountability	5 years (anonymize user names)	7 years (delete)
Audit Logs (Incidents)	10 years	Legal defense, regulatory compliance	7 years (anonymize)	12 years (delete)
IncidentType Catalog	Indefinite	System configuration (no personal data)	N/A	Never (unless deprecated)
TaskCategory Catalog	Indefinite	System configuration (no personal data)	N/A	Never (unless deprecated)

7.2. Justifications

FAL Convention Data Retention Requirements:

FAL Convention Article 2.11.1: Documents presented at arrival can be reused at departure if vessel remains in same port system.

FAL Convention Section 2.F (Articles 2.6.3, 2.13): Documents may be retained for **14 days** for vessels on established schedules without changes in crew/passengers.

Implication for Retention Policy:

- **Active VVN Data:** Retain while vessel is in port system + 14 days (FAL Convention compliance)
- **Historical VVN Data:** After active period, maritime law and insurance requirements apply (7-10 years)
- **Crew/Passenger Lists:** Minimum retention not specified by FAL for long-term storage, but health authorities may require retention for disease tracking (FAL Convention Article 4.8 - public health surveillance)

Recommended Retention Period (VVN Personal Data):

- **Active Period:** During vessel visit + 14 days (FAL Convention Articles 2.6.3, 2.11.1)
- **Archival Period:** 7 years (maritime insurance claims, safety audits, IMO regulations)
- **Anonymization:** After 7 years, anonymize crew/passenger CitizenIds - retain aggregated statistics only for port efficiency analysis

7-10 Year Retention for VVEs and Incidents:

- **IMO Regulations:** Maritime incident records required for safety audits (5-10 years)
- **Insurance Claims:** Claims can be filed up to 6 years after incident (Portuguese Civil Code)
- **Legal Defense:** Statute of limitations for civil liability: 10 years (Portuguese law)
- **FAL Convention Emergency Records:** Articles 2.17-2.24 (medical emergency incidents) - no specific retention period, but 7-10 years aligns with safety audit requirements

3 Year Retention for Operation Plans:

- Operational efficiency analysis typically looks back 1-3 years

- Dispute resolution for cargo handling complaints: 2-3 years

2 Year Retention for Complementary Tasks:

- Short-term operational data only
- No legal requirement for long-term retention

7.3. Implementation of Retention Policy

Scheduled Jobs:

```
// Runs monthly on 1st day at 3 AM
cron.schedule('0 3 1 * *', async () => {
  const now = new Date();

  // Anonymize 3-year-old operation plans
  const threeYearsAgo = new Date();
  threeYearsAgo.setFullYear(now.getFullYear() - 3);
  await operationPlanRepo.anonymizeOlderThan(threeYearsAgo);

  // Delete 2-year-old tasks
  const twoYearsAgo = new Date();
  twoYearsAgo.setFullYear(now.getFullYear() - 2);
  await taskRepo.deleteOlderThan(twoYearsAgo);

  // Anonymize 7-year-old VVEs
  const sevenYearsAgo = new Date();
  sevenYearsAgo.setFullYear(now.getFullYear() - 7);
  await vveRepo.anonymizeStaffIds(sevenYearsAgo);

  // Log retention actions
  await auditLog.create({
    action: 'AUTOMATED_RETENTION_POLICY_EXECUTED',
    timestamp: now,
    details: 'Monthly data retention and anonymization completed'
  });
});
```

---

8. Breach Response Procedures (Updated for Dual-System Architecture)

8.1. Sprint C Breach Scenarios

Scenario 1: MongoDB Data Breach

- **What:** Attacker gains access to OEM MongoDB database
- **Exposed Data:** User IDs, staff IDs, incident descriptions, external entity names, audit logs
- **Severity Assessment:**
  - **High Risk** if: External entity names include individuals, incident descriptions contain sensitive data

- **Medium Risk** if: Only user IDs exposed (still linkable to Core Backend data)

Scenario 2: Core-to-OEM API Compromise

- **What:** Attacker intercepts API calls between Core Backend and OEM Backend
- **Exposed Data:** VVN details (including crew CitizenId), user profiles, staff details
- **Severity Assessment: Critical** (highly sensitive personal data exposed)

Scenario 3: JWT Token Theft (Cross-System Impact)

- **What:** XSS attack steals JWT token from browser localStorage
- **Exposed Data:** Token grants access to BOTH Core and OEM backends
- **Severity Assessment: High to Critical** depending on user role (Admin tokens = worst case)

8.2. Breach Detection Mechanisms

For OEM Backend:

1. **Failed Authentication Monitoring:**

```
// Log failed MongoDB authentication attempts
if (mongoError.code === 18) {  // Authentication failed
  await securityAlert.send('MONGODB_AUTH_FAILURE', { ip, timestamp });
}
```

2. **Unusual API Access Patterns:**
   - Multiple IPs using same JWT token
   - Excessive data queries (e.g., 1000+ incidents fetched in 1 minute)
   - Access from unusual geographic locations
3. **Database Audit Logging:**
   - Enable MongoDB audit log (Enterprise) or application-level logging
   - Monitor: Unauthorized schema changes, bulk exports, admin account creation

8.3. Breach Response Procedure (72-Hour Timeline)

Hour 0-4: Detection and Containment

1. Security team detects breach (automated alert or manual discovery)
2. **Immediate Actions:**
   - Isolate compromised system (firewall rules, network segmentation)
   - Revoke all active JWT tokens (if token breach)
   - Change MongoDB admin password (if database breach)
   - Preserve evidence (logs, database snapshots)
3. Activate incident response team

Hour 4-12: Assessment

1. Determine scope:
    - Which data was accessed/exfiltrated?
    - How many data subjects affected?
    - Was data encrypted?
2. Classify severity:
    - **High:** Special category data (CitizenId, health data), large scale (>1000 subjects)
    - **Medium:** User IDs, contact info, moderate scale (100-1000 subjects)
    - **Low:** System logs, aggregated data, small scale (<100 subjects)

Hour 12-24: Notification Preparation

1. Draft breach notification for CNPD (Portuguese supervisory authority)
2. Draft notification for affected data subjects (if required)
3. Coordinate between Core Backend and OEM Backend teams (both systems may be affected)

Hour 24-72: Notification

1. **Notify CNPD within 72 hours** (Article 33):
    - Nature of breach
    - Categories and approximate number of data subjects affected
    - Contact point (DPO)
    - Likely consequences
    - Measures taken or proposed
2. **Notify data subjects** (Article 34) if:
    - High risk to rights and freedoms (e.g., CitizenId exposed)
    - Cannot demonstrate data was encrypted or unintelligible
    - **Exception:** Public communication may be used if individual notification is disproportionate

Hour 72+: Remediation and Lessons Learned

1. Implement fixes (e.g., enable MongoDB encryption, patch XSS vulnerability)
2. Conduct post-incident review
3. Update security policies and procedures
4. Retrain staff on breach prevention

---

9. Compliance Roadmap for Sprint C

9.1. Timeline Overview

```
Sprint C Completion (Now - January 2026)
│
├── ☑ COMPLETED (US 4.5.x - GDPR Data Subject Rights)
│   ├── Privacy Policy Management (US 4.5.1, 4.5.2)
│   ├── User Data Rights / SAR (US 4.5.3)
│   └── Non-User Data Requests (US 4.5.4)
│
├── Immediate (0-4 weeks) - CRITICAL GAPS
│   ├── Secure MongoDB (authentication, firewall)
│   ├── Remove cached userNames from audit logs
```

```
│   └── Operator training on GDPR-compliant incident reporting
│
├── Short-Term (1-3 months) - HIGH PRIORITY
│   ├── Enable MongoDB encryption at rest
│   ├── OEM Backend integration for complete SAR data export
│   ├── Email notifications for data request status
│   ├── Automated PII detection in free-text fields
│   ├── Service-level agreement (Joint Controller Agreement)
│   └── Anonymized incident reports for third parties
│
├── Medium-Term (3-6 months) - MEDIUM PRIORITY
│   ├── Coordinated data retention and deletion
│   ├── Token revocation mechanism (Redis blacklist)
│   ├── Conduct DPIA for incident management
│   └── Cross-service audit logging improvements
│
└── Long-Term (6-12 months) - LOW PRIORITY
    ├── Field-level encryption for sensitive fields
    ├── Centralized audit logging (ELK stack)
    ├── Privacy-enhancing technologies (differential privacy for analytics)
    └── Continuous compliance monitoring and audits
```

9.2. Resource Allocation

Phase 1 (Immediate): 2 developers, 1 legal advisor, 1 trainer → 4 weeks
Phase 2 (Short-term): 2-3 developers, 1 legal advisor → 3 months
Phase 3 (Medium-term): 2 developers, 1 data protection consultant → 3 months
Phase 4 (Long-term): 1-2 developers, ongoing → 6 months

Total Estimated Cost:

- Development effort: 60-80 developer days (€40,000-€60,000)
- Legal consultation: 30-40 hours (€6,000-€10,000)
- MongoDB Enterprise license or Atlas (€5,000-€15,000/year)
- Total: **€50,000-€85,000**

Risk of Non-Compliance:

- Potential GDPR fines: €10-€20 million or 2-4% of annual turnover
- Reputational damage
- Loss of business (shipping agents may refuse to use non-compliant systems)

ROI Analysis: Compliance investment represents 0.25-0.5% of potential fine → **Highly cost-effective**

---

10. Sprint C-Specific GDPR Conclusions

10.1. Summary of Sprint C GDPR Posture

Positive Aspects: ☑ OEM Backend minimizes data duplication (user IDs only, not full profiles)
☑ Clean Architecture enables future privacy-enhancing modifications

☑ MongoDB supports encryption and field-level encryption (not yet enabled)
☑ Domain-driven design enforces business rules (audit logging built into entities)
☑ RESTful APIs enable access control and rate limiting (future implementation)
☑ **NEW: Privacy Policy System (US 4.5.1, 4.5.2)** - Version management, public display, user consent tracking
☑ **NEW: Subject Access Request mechanism (US 4.5.3)** - Data export (JSON/PDF), rectification, deletion
☑ **NEW: Non-User Data Rights (US 4.5.4)** - External data subjects can exercise GDPR rights
☑ **NEW: Data Request Management** - Status tracking, admin processing, request history

Remaining Gaps: ⚠ MongoDB has no authentication, no encryption at rest (security configuration pending)
⚠ Personal data cached in audit logs (`userName` field violates data minimization)
⚠ Free-text fields in incidents may contain unstructured PII
⚠ External entity names may include personal data (third-party data without legal basis)
⚠ No coordinated data retention policy across Core + OEM
⚠ No formal agreement between Core Backend and OEM Backend (joint controller gap)
⚠ OEM Backend data not yet included in SAR exports (future enhancement)
⚠ Email notifications for data request status not implemented

10.2. Inherited Gaps from Sprint A and Sprint B

Sprint C does not address previous critical gaps:

- ❌ CitizenId encryption at rest in Core Backend SQL Server (Sprint A issue)
- ❌ JWT token storage in localStorage (Sprint B XSS vulnerability)
- ❌ No Data Processing Agreement with Google OAuth (Sprint B third-party processor)
- ☑ **RESOLVED: Subject Access Request mechanism implemented (US 4.5.3)** - Users can export, request rectification, and deletion
- ☑ **RESOLVED: Privacy Policy implemented (US 4.5.1, 4.5.2)** - Display and consent tracking
- ☑ **RESOLVED: Non-User Data Requests (US 4.5.4)** - External data subjects can submit requests
- ⚠ Breach detection and notification procedures (Sprint A/B gap, now affects both systems)

Recommendation: Future sprints should prioritize **OEM Backend integration** for complete data export and **email notifications** for data request status updates.

10.3. Unique Sprint C GDPR Challenges

Challenge 1: Distributed Personal Data Architecture

- User personal data split across SQL Server (Core) and MongoDB (OEM)
- Data subject rights requests require querying both databases
- Deletion/anonymization must be coordinated

Challenge 2: External Entity Data Collection

- Incidents may involve third parties (Customs, Fire Department, Police)
- Unclear controller/processor roles
- No consent or formal data sharing agreements

Challenge 3: Dual-Purpose Staff Data

- Staff assignments tracked for safety/operations (legal obligation)
- Same data enables performance evaluation (legitimate interest or consent required)
- Purpose creep risk

Challenge 4: NoSQL Data Model Flexibility

- MongoDB allows flexible schemas (easy to add new fields)
- Risk: Developers add personal data fields without GDPR assessment
- Need: Schema governance and privacy-by-design review process

10.4. Path to Full GDPR Compliance

To achieve full GDPR compliance for the Port Management System (Sprint A + B + C):

Tier 1: Prevent Immediate Fines

1. Secure MongoDB database (authentication, encryption, firewall)
2. Remove cached personal data from OEM audit logs
3. Encrypt CitizenId in Core Backend SQL Server (Sprint A gap)
4. Migrate JWT tokens from localStorage to httpOnly cookies (Sprint B gap)
5. Document data retention policies
6. Establish Joint Controller Agreement between Core and OEM

Tier 1 - ☑ COMPLETED (Sprint C US 4.5.x): 7. ☑ Privacy Policy display and versioning (US 4.5.1) 8. ☑ User consent/acknowledgment tracking (US 4.5.2) 9. ☑ Subject Access Request mechanism - export, rectification, deletion (US 4.5.3) 10. ☑ Non-user data request handling (US 4.5.4)

Tier 2: Establish Operational Compliance 11. OEM Backend integration for complete SAR data export 12. Email notifications for data request status changes 13. Automated data retention and anonymization 14. Token revocation mechanism 15. Automated PII detection for incident free-text fields 16. Breach detection and notification procedures 17. DPAs with Google OAuth and Gmail SMTP

Tier 3: Advanced Compliance and Continuous Improvement 18. Centralized audit logging (ELK stack) 19. Field-level encryption for sensitive data 20. Privacy-enhancing technologies (differential privacy, synthetic data for testing) 21. Regular DPIAs for new features 22. Staff training program on data protection 23. Annual compliance audits

Expected Benefit:

- Regulatory compliance (avoid high fines)
- Customer trust (data protection as competitive advantage)
- Operational resilience (robust security and audit capabilities)
- Legal defensibility (documented compliance posture)

---

11. References

11.1. GDPR and Data Protection Framework

1. **Regulation (EU) 2016/679 (GDPR)**
   European Parliament and Council. (2016). *General Data Protection Regulation*.

Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj

2. **European Data Protection Board (EDPB)**
   EDPB. (2026). *Guidelines and Recommendations on GDPR Compliance*.
   Available at: https://edpb.europa.eu/our-work-tools/general-guidance_en

3. **Portuguese Supervisory Authority (CNPD)**
   Comissão Nacional de Proteção de Dados. (2026). *Data Protection Authority Portal*.
   Available at: https://www.cnpd.pt/

4. **EDPB Guidelines 4/2019 on Article 25**
   European Data Protection Board. (2020). *Guidelines on Data Protection by Design and by Default*.
   Available at: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-42019-article-25-data-protection-design-and_en

5. **EDPB Guidelines 07/2020 on Controller-Processor Relationship**
   European Data Protection Board. (2021). *Guidelines on the concepts of controller and processor in the GDPR*.
   Available at: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-072020-concepts-controller-and-processor-gdpr_en

6. **EDPB Guidelines 05/2020 on Consent**
   European Data Protection Board. (2020). *Guidelines 05/2020 on consent under Regulation 2016/679*.
   Available at: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-052020-consent-under-regulation-2016679_en

7. **EDPB Guidelines 8/2020 on Targeting of Social Media Users**
   European Data Protection Board. (2021). *Guidelines on the targeting of social media users*.
   Available at: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-82020-targeting-social-media-users_en

11.2. International Data Transfers

8. **EU-US Data Privacy Framework**
   European Commission. (2023). *Adequacy Decision for the EU-US Data Privacy Framework*.
   Available at: https://www.dataprivacyframework.gov/

9. **Standard Contractual Clauses (SCCs)**
   European Commission. (2021). *Implementing Decision on standard contractual clauses for international data transfers*.
   Available at: https://ec.europa.eu/info/law/law-topic/data-protection/international-dimension-data-protection/standard-contractual-clauses-scc_en

11.3. Maritime and Port Security Regulations

10. **International Ship and Port Facility Security (ISPS) Code**
    International Maritime Organization (IMO). (2003). *ISPS Code - Part A and Part B*.
    Available at: https://www.imo.org/en/OurWork/Security/Pages/SOLAS-XI2%20ISPS%20Code.aspx

11. **EU Regulation 725/2004**
European Parliament and Council. (2004). *Regulation on enhancing ship and port facility security*. Available at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32004R0725

12. **IMO FAL Convention** International Maritime Organization. (1965, amended 2024). *Convention on Facilitation of International Maritime Traffic*. Available at: https://www.imo.org/en/About/Conventions/Pages/Convention-on-Facilitation-of-International-Maritime-Traffic-(FAL).aspx

12.1. **Decreto n.º 13/1990 - FAL Convention Portuguese Ratification** Portuguese Republic. (1990). *Convenção sobre Facilitação do Tráfego Marítimo Internacional, de 1965*. Diário da República, Série I-A, 1990-04-19.

```
 **Key Provisions for GDPR Compliance:**
 - Article 1.1: Data minimization requirement for port authorities ("informações
indispensáveis" - essential information only)
 - Article 2.6: Crew list mandatory data elements (name, nationality, rank,
date/place of birth, identity document)
 - Article 2.7: Passenger list mandatory data elements (name, nationality,
date/place of birth, ports of embarkation/disembarkation)
 - Article 2.9: Health declaration requirements (special category data under GDPR
Article 9)
 - Article 4: International cooperation with external entities (Customs, Health
Authorities, Coast Guard)
 - Articles 2.6.3, 2.11.1, 2.13: Document retention (14 days for scheduled
vessels)
 - Articles 2.17-2.24: Medical emergency procedures
 - Articles 5.11-5.12: Emergency assistance coordination
```

11.4. Third-Party Service Providers (Sprint B + Sprint C)

13. **Google Privacy Policy**
Google LLC. (2026). *Privacy Policy*.
Available at: https://policies.google.com/privacy

14. **MongoDB Security Documentation**
MongoDB, Inc. (2026). *MongoDB Security Checklist*.
Available at: https://www.mongodb.com/docs/manual/administration/security-checklist/

15. **MongoDB Encryption at Rest**
MongoDB, Inc. (2026). *Encryption at Rest*.
Available at: https://www.mongodb.com/docs/manual/core/security-encryption-at-rest/

16. **MongoDB Client-Side Field Level Encryption**
MongoDB, Inc. (2026). *Client-Side Field Level Encryption*.
Available at: https://www.mongodb.com/docs/manual/core/csfle/

17. **MongoDB Atlas Data Privacy and Security**
MongoDB, Inc. (2026). *MongoDB Atlas Privacy*.

Available at: https://www.mongodb.com/cloud/trust

11.5. Security Standards and Best Practices

18. **OWASP Top 10 Web Application Security Risks**
OWASP Foundation. (2021). *OWASP Top 10:2021*.
Available at: https://owasp.org/www-project-top-ten/

19. **NIST Cybersecurity Framework**
National Institute of Standards and Technology. (2018). *Framework for Improving Critical Infrastructure Cybersecurity*.
Available at: https://www.nist.gov/cyberframework

20. **ISO/IEC 27001:2022**
International Organization for Standardization. (2022). *Information security management systems - Requirements*.
Available at: https://www.iso.org/standard/27001

21. **NIST Special Publication 800-53 (Security Controls)**
National Institute of Standards and Technology. (2020). *Security and Privacy Controls for Information Systems and Organizations*.
Available at: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

11.6. Technical Documentation

22. **JWT (JSON Web Token) Specification - RFC 7519**
Internet Engineering Task Force (IETF). (2015). *JSON Web Token (JWT)*.
Available at: https://datatracker.ietf.org/doc/html/rfc7519

23. **OAuth 2.0 Authorization Framework - RFC 6749**
Internet Engineering Task Force (IETF). (2012). *The OAuth 2.0 Authorization Framework*.
Available at: https://datatracker.ietf.org/doc/html/rfc6749

24. **RESTful API Security Best Practices**
OWASP Foundation. (2024). *REST Security Cheat Sheet*.
Available at: https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

25. **Node.js Security Best Practices**
Node.js Foundation. (2026). *Security Best Practices*.
Available at: https://nodejs.org/en/docs/guides/security/

11.7. NoSQL and MongoDB-Specific Security

26. **MongoDB Security Architecture**
MongoDB, Inc. (2026). *Security Architecture White Paper*.
Available at: https://www.mongodb.com/collateral/mongodb-security-architecture

27. **OWASP NoSQL Injection**
OWASP Foundation. (2024). *NoSQL Injection*.
Available at: https://owasp.org/www-community/attacks/NoSQL_injection

28. **MongoDB Authentication Mechanisms**
MongoDB, Inc. (2026). *Authentication*.
Available at: https://www.mongodb.com/docs/manual/core/authentication/

11.8. Microservices and Distributed Systems Security

29. **NIST Special Publication 800-204 (Microservices Security)**
National Institute of Standards and Technology. (2019). *Security Strategies for Microservices-based Application Systems*.
Available at: https://csrc.nist.gov/publications/detail/sp/800-204/final

30. **Service Mesh Security (Istio Documentation)**
Istio Authors. (2026). *Security Concepts*.
Available at: https://istio.io/latest/docs/concepts/security/

11.9. Data Retention and Archival

31. **IMO Records and Reporting Requirements**
International Maritime Organization. (2024). *Mandatory Audit Scheme - Records Retention*.
Available at: https://www.imo.org/en/OurWork/MSAS/Pages/default.aspx

32. **Portuguese Civil Code (Código Civil)**
Portuguese Republic. (1966, amended 2024). *Civil Code - Statute of Limitations*.
Available at: https://www.pgdlisboa.pt/leis/lei_mostra_articulado.php?nid=775&tabela=leis

---

Document Control

Version History:

Version	Date	Author	Changes
2.0	January 3, 2026	Grupo 05	Second version of Sprint C report
2.1	January 4, 2026	Grupo 05	Updated to reflect US 4.5.1, 4.5.2, 4.5.3, 4.5.4 implementation: Privacy Policy, User Data Rights (SAR), Non-User Data Requests

Distribution:

- Development Team (Core Backend + OEM Backend)
- Information Security Team
- Legal and Compliance Department
- Port Authority Management
- CNPD (Portuguese Supervisory Authority) - upon request