Security & Data Protection Plan

BORG | Industrial Digitalization

Bram - BORG

March 2026

Executive Summary

This document defines the security framework for BORG (ai-borg.be), an industrial digitalization company that builds AI-driven software for the manufacturing sector.

What BORG stores: Application-level data only. User accounts (names, email addresses, hashed passwords), roles, permissions, and in safety-critical apps, audit logs and photo evidence. We don't run servers, manage firewalls, or operate infrastructure. We build applications, and we secure the data those applications handle.

Where it lives: All app data is stored in Supabase, an enterprise-grade open-source platform running on AWS in the EU (Frankfurt). Supabase manages infrastructure security: encryption, authentication, backups, patching, and uptime. BORG manages application-layer security: data isolation (Row Level Security), secure coding, API key discipline, and GDPR compliance.

AI and your data: By default, no client data is sent to any external AI service. AI powers our development process, not your data pipeline. If a client wants AI to work with their operational data, we offer two options: a local AI model (maximum privacy, higher cost) or an external AI provider (cost-effective, client explicitly consents). The client decides. This is documented in the contract and the Data Processing Agreement.

Who Handles What

Security Control	Owner	What This Means
Encryption at rest (AES-256)	Supabase	Your database is encrypted automatically
Encryption in transit (TLS 1.2+)	Supabase	Every connection is secured
Automated daily backups	Supabase	Point-in-time recovery, EU-stored
Infrastructure security	Supabase	SOC 2 Type II certified, managed security team
DDoS protection and patching	Supabase	Handled without BORG involvement
User authentication	Supabase	Supabase Auth: bcrypt hashing, sessions, MFA

Security Control	Owner	What This Means
Row Level Security (RLS)	BORG	Database-level client data isolation
API key management	BORG	Secret keys never in client-side code
Input validation	BORG	Server-side validation, parameterized queries
Secure coding	BORG	OWASP-aligned, tested before every deployment
Deployment verification	BORG	10-point security checklist
Dependency management	BORG	Vulnerability scanning, 48-hour critical patches
GDPR compliance	BORG	DPA, processing register, breach procedures
AI data governance	BORG	Client chooses AI model: local, external, or none

AI Data Governance

BORG uses AI to build software faster and deliver better results. But how AI interacts with *your* data is your decision, not ours. Every BORG project starts with a clear AI data classification.

The Three Options

Option 1: No AI On Client Data (Default)

This is the standard for every project unless the client requests otherwise.

- AI powers BORG's development process: code generation, architecture, testing
- No client operational data is sent to any external AI service
- No client data is used to train, fine-tune, or improve any AI model
- All client data stays within EU-hosted Supabase infrastructure

This option applies automatically. No additional documentation or consent required.

Option 2: Local AI Model (Maximum Privacy)

For clients who want AI features that work with their data but require full data privacy.

- AI model runs on dedicated local hardware (BORG-managed or client-managed)
- Client data never leaves the controlled environment
- No third-party AI provider involved
- Higher project cost due to dedicated compute resources
- Examples: AI-powered search over client documents, predictive maintenance analysis, natural language queries against operational databases

This option requires: documented scope of AI access in the DPA, client sign-off on the specific data categories the AI model can access.

Option 3: External AI Provider (Cost-Effective)

For clients who want AI features and prioritize cost-effectiveness over full data isolation.

- Client data is sent to an external AI provider (e.g., Anthropic Claude API) for processing
- The AI provider is documented as a sub-processor in the Data Processing Agreement
- Client explicitly consents to this data flow before any data is transmitted
- BORG selects providers with strong data handling policies (no training on client data, data deletion after processing)
- Lower cost than local models, faster to implement, access to the most capable AI models available

This option requires: explicit client consent, the AI provider added to the sub-processor register, documented data categories and processing purposes in the DPA.

AI Governance in the Contract

Every client contract specifies which AI option applies. The Data Processing Agreement includes:

- Which option was selected (1, 2, or 3)
- If option 2 or 3: exactly which data categories the AI model can access
- If option 3: which external provider is used and their data handling commitments
- The client's right to change their AI preference with written notice

This is not a default buried in terms and conditions. It's a conversation we have during the project scoping phase, and the client's choice is documented clearly.

Platform Security (Supabase)

We chose Supabase because it gives every app enterprise-grade infrastructure from day one. This section documents what the platform handles so there is no ambiguity about where BORG's responsibility begins and ends.

Infrastructure

- All projects provisioned in the EU (Frankfurt, AWS eu-central-1)
- SOC 2 Type II certified with a dedicated security team
- Database connections encrypted with TLS 1.2+; data encrypted at rest with AES-256
- DDoS protection, network security, and OS patching all managed by Supabase

BORG does not manage, configure, or have access to the underlying servers, network, or CDN layer. That is entirely Supabase's domain.

Authentication (Supabase Auth)

Every BORG application uses Supabase Auth. We don't roll our own authentication.

- Passwords hashed with bcrypt (industry standard, one-way)
- Secure session management with configurable expiry
- Multi-factor authentication (MFA/2FA) available
- OAuth integration (Google, Microsoft) for SSO
- Built-in rate limiting on login endpoints
- Email confirmation and password reset flows included

This is the same authentication infrastructure used by thousands of production applications. We didn't build it, and that's the point. Reinventing auth is how breaches happen.

Backups

Supabase provides automated daily backups with point-in-time recovery on Pro plans. All BORG client projects use plans that include this. Backups are encrypted and stay in the EU.

BORG does not run separate backups of app data. Supabase's managed backup service covers availability and recovery.

Application Security (BORG)

This is where our work actually lives. Everything above this section is Supabase's job. Everything below is ours.

Row Level Security (RLS)

This is the single most important security control we manage.

Row Level Security is a Supabase/PostgreSQL feature that enforces data access rules at the database level. Even if our application code had a bug, the database itself would prevent unauthorized access. It's a safety net built into the data layer.

- RLS is enabled on every table that stores client or user data
- Policies enforce that users can only see and modify data belonging to their own organization
- Before every deployment, we test cross-tenant access: can User A see Client B's data? If yes, the deployment does not proceed.
- No table containing personal or client data exists without RLS policies

If RLS is misconfigured, one client's user could see another client's data. That's why we test it explicitly every time.

API Key Management

Supabase provides two API keys. How you handle them determines whether your RLS actually matters.

Key Type	What It Does	Our Policy
anon (public)	Respects RLS policies	This is the only key that goes in frontend code. All data access is governed by RLS.
service_role (secret)	Bypasses all RLS	NEVER in client-side code. Server-side only. Stored in .env files, excluded from git.

If the service_role key ends up in frontend code, your RLS is bypassed and anyone with browser dev tools can access everything. We treat this as a critical rule:

- Service role keys live in .env files with restricted permissions
- .env files are always in .gitignore
- We scan git history for accidentally committed keys
- If a key is compromised, it's rotated immediately

Secure Coding Standards

What	Why	How
Input Validation	Prevent injection attacks	Server-side validation, parameterized queries via Supabase client, HTML escaping
XSS Prevention	Stop script injection	Content Security Policy headers, React/Next.js auto-escaping
CSRF Protection	Prevent unauthorized actions	SameSite cookies, Origin/Referer verification
API Security	Lock down every endpoint	Supabase Auth tokens validated server-side, restrictive CORS
File Uploads	Prevent malicious files	Supabase Storage with bucket policies, MIME type whitelist, size limits
Error Handling	Don't leak system info	Generic error pages in production, detailed errors logged server-side only
Logging	Track problems, not secrets	Auth attempts and failures logged; passwords and

What	Why	How
		tokens never logged

Safety-Critical Application Controls

Apps involving workplace safety (LoTO procedures, maintenance sign-offs) have additional requirements because the consequences of unauthorized access aren't just data exposure — they're physical safety risks.

- Individual accounts only. No shared logins, no generic team accounts.
- Every action is logged: user ID, timestamp, device identifier, action taken
- Audit logs are immutable. Nobody can edit or delete them after the fact.
- Photo evidence attached to sign-offs cannot be replaced or deleted after submission
- Role-based access control: Operators, Supervisors, and Administrators have distinct permissions

Deployment Checklist

Every production deployment passes these checks. No exceptions.

#	Check	Area
1	Environment variables set; no service_role keys in client-side code	Secrets
2	HTTPS enforced on all endpoints	Transport
3	Security headers configured (HSTS, CSP, X-Frame-Options)	Transport
4	Supabase Auth tested for all user roles	Access
5	RLS tested with cross-tenant verification	Data isolation
6	Dependency audit: no critical vulnerabilities	Dependencies
7	Database migrations tested on staging first	Data
8	Error pages show no stack traces or system info	Disclosure
9	CORS configured restrictively (no wildcard origins)	Access
10	File upload restrictions tested	Input

GDPR Compliance

BORG applications store personal data: names, email addresses, hashed passwords, and in safety apps, action logs tied to individuals. That makes us a data processor under GDPR. Here's how we handle it.

What We Actually Store

Data	Examples	Why
User accounts	Name, email, hashed password	People need to log in
Roles	User role, tenant/organization ID	Access control and data isolation
Audit logs (safety apps)	User ID, timestamp, action	Compliance and traceability
Photos (safety apps)	Sign-off photos	Proof that safety procedures were followed

Client operational data (stock levels, site info, schedules) may also live in the app, but that's the client's data, not personal data under GDPR. It gets the same security treatment regardless.

Data Processing Agreement (DPA)

Every client contract includes a DPA. It covers:

1. **Scope:** What data we process, why, and for how long
2. **Instructions:** We only process data as the client directs
3. **Security:** Reference to the technical measures in this document
4. **Sub-processors:** Supabase (AWS EU) listed; AI providers listed if option 2 or 3 selected
5. **AI data classification:** Which AI option the client chose, what data AI can access
6. **Data subject rights:** We help fulfill access, correction, and deletion requests
7. **Breach notification:** Client hears from us within 24 hours
8. **Data return:** When the contract ends, data is returned and deleted within 30 days
9. **Audit:** Clients can audit our compliance with reasonable notice

Breach Notification

When	What Happens
0-4 hours	Contain the issue. Figure out what happened, what data was affected, and which clients are impacted. Preserve evidence.
4-24 hours	Notify affected clients: what happened, what data, what we're doing about it.
24-72 hours	If required: notify Belgian Data Protection Authority (GBA/APD) with full details and remediation plan.
72+ hours	Root cause analysis. Fix the problem. Update

When	What Happens
	security measures. Deliver final report to clients.

Client Contract Security Package

Every BORG client gets these documents with their contract:

Document	What It Is
Data Processing Agreement (DPA)	GDPR Article 28 compliant processing terms, including AI data classification
Security & Data Protection Plan	This document
AI Data Governance Agreement	Documents the client's chosen AI option and data scope
Sub-Processor Register	Every third-party service that touches client data

Appendix A: Sub-Processor Register

Sub-Processor	What They Do	Where	When Applicable
Supabase Inc.	Database, authentication, file storage, backups	EU (Frankfurt, AWS)	All projects
Vercel (if applicable)	Application hosting and edge deployment	EU region available	When used for hosting
Anthropic (Claude API)	AI processing of client operational data	US/EU	Only if client selects AI Option 3
OpenAI (API)	AI processing of client operational data	US/EU	Only if client selects AI Option 3

Updated when sub-processors change. Clients are notified as required by the DPA. AI providers are only added when the client explicitly opts for Option 3.

Appendix B: AI Data Governance Statement

BORG AI Data Governance Commitment

BORG uses AI to build better software faster. How AI interacts with your data is your decision. We offer three clear options:

Default (Option 1): No client data is sent to any external AI service. AI powers our development process only. All client data stays within EU-hosted Supabase infrastructure.

Local AI (Option 2): For clients who want AI features on their data with maximum privacy. The AI model runs on dedicated local hardware. Data never leaves the controlled environment. Higher project cost due to compute requirements.

External AI (Option 3): For clients who want AI features at lower cost. Data is processed by an external AI provider (e.g., Anthropic Claude API). The client explicitly consents, and the provider is documented as a sub-processor.

Regardless of which option you choose:

- No client data is ever used to train, fine-tune, or improve any AI model
- The selected option is documented in your contract and DPA
- You can change your AI preference at any time with written notice
- BORG will always explain the trade-offs honestly so you can make an informed decision

Document Control

Version	Date	Author	Changes
1.0	March 2026	Bram, BORG	Initial framework

Reviewed quarterly. Next review: June 2026.