GDPR Compliance for AI Agents and Infrastructure Access

Control what engineers and AI agents can see, run, and extract from production systems. Enforce GDPR at the protocol layer.

- Five GDPR articles mapped to infrastructure controls
- Protocol-level PII masking with zero configuration
- Immutable audit trails for DPA evidence requests
- Zero stored credentials across all access paths

CONTENTS

THE CORE PROBLEM

GDPR regulates runtime access, not just stored data

When an engineer queries a production database, that action constitutes personal data processing under GDPR. Most organizations treat infrastructure access as an operations problem. The regulation treats it as a regulated activity.

Security teams deploy encryption at rest and TLS in transit. They configure IAM policies and rotate secrets. None of those controls govern what happens after an engineer opens a database connection and runs a SELECT against a table containing personal data.

1 Requester

An engineer, AI agent, or pipeline queries production data. Today this means a slow approval chain or no controls at all.

2 In-Session Control

Hoop.dev intercepts at the protocol layer. Masking, firewall rules, and JIT reviews happen inline. No credentials stored.

3 Compliant by Default

Access granted in seconds. PII redacted. Commands logged. Audit trail generated. No bottleneck. No blind spot.

WHERE THE EXPOSURE LIVES

Database Queries

Engineers run ad-hoc SQL against production tables containing personal data. No field-level access control exists at the query layer.

Server Access

SSH sessions to production hosts expose log files, configuration data, and application state containing PII.

Kubernetes Operations

kubectl exec and log access expose container-level data, including environment variables with connection strings.

Application Debugging

Incident response requires direct data inspection. Engineers access unmasked records under time pressure without audit trails.

QUICK REFERENCE

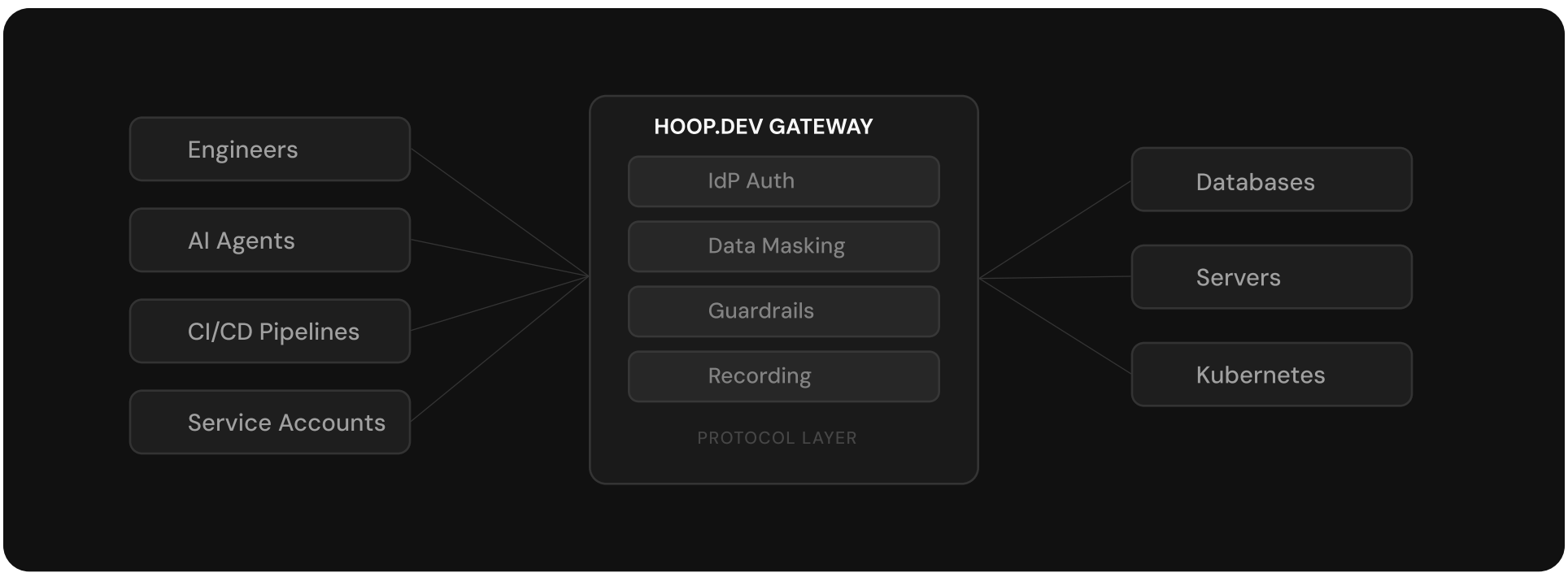
GDPR article-to-control mapping

ARTICLE	REQUIREMENT	HOOP.DEV CONTROL	TIER
Art. 25	Data protection by design and by default	AI data masking (150+ PII types)	DIRECT
Art. 32	Appropriate technical measures for security	SSO gateway, pseudonymization	DIRECT
Art. 5(2)	Demonstrate compliance (accountability)	Immutable session recording	DIRECT
Art. 30	Records of processing activities	Command-level audit logs	DIRECT
Art. 5(1)(f)	Integrity, confidentiality, least privilege	Zero stored credentials, JIT access	DIRECT
Art. 15-20	Data subject access and erasure requests	Queryable access event history	SUPPORTING

The key insight: GDPR compliance at the infrastructure layer is not about building new processes. It is about placing a protocol-aware proxy between identities and data. Every article above traces back to what that proxy can see, mask, log, and enforce.

IN-SESSION CONTROL FOR EVERY IDENTITY

One gateway for all identities, down to the command level



EVERY ACTION FLOWS THROUGH THE GATEWAY

1. **Query initiation**
   Command sent through the hoop.dev proxy
2. **SSO authentication**
   OpenID Connect confirms identity
3. **Protocol interception**
   Gateway parses at wire protocol level
4. **Policy evaluation**
   Guardrails check rules. Block or gate.
5. **JIT credential retrieval**
   Agent pulls creds from secrets manager
6. **Execution**
   Query runs against the target system
7. **Response masking**
   ML masks 150+ PII types before delivery
8. **Audit capture**
   Command, identity, and redaction logged

DIRECT COVERAGE: WHERE HOOP.DEV SATISFIES THE REQUIREMENT

Five articles, one gateway

Each article below maps to a specific control enforced at the protocol layer. These are not adjacent capabilities. They are direct fits: the gateway generates the primary evidence each article requires.

ARTICLE 25

Data Protection by Design and by Default

Hoop.dev masks personal data in real time at the protocol layer, before results reach the engineer's screen. No rules to configure. No policies to maintain. Sensitive fields are redacted automatically across every database and server connection.

ARTICLE 32

Appropriate Technical Measures

All access is brokered through an SSO-authenticated gateway. Personal data is automatically pseudonymized in query outputs. Raw credentials are never exposed to users. Engineers do their jobs without touching unmasked personal data.

ARTICLES 5(2) + 30

Processing Records and Accountability

Every access event is logged with full attribution: who accessed what, which commands were executed, what data was returned, and when. Logs are immutable and retained per configurable policy. When a DPA requests evidence, query the logs directly.

ARTICLE 5(1)(F)

Least Privilege and Access Limitation

Hoop.dev never stores credentials or accesses private keys. The gateway brokers access using secrets held in the customer's own secret manager. Users authenticate via SSO. The regulation prohibits vendors from touching private keys. This architecture makes it structurally impossible.

Masking strategies

Sensitive data gets caught in transit, not after the fact. ML-powered detection masks PII at the protocol layer before the response reaches the requester.

STRATEGY	EXAMPLE	USE CASE
Full redaction	[REDACTED]	Passwords, auth tokens, encryption keys
Partial masking	****-****-****-1234	Credit cards, SSNs, phone numbers
Hashing	SHA256(value)	User IDs, session identifiers
Format preserving	Structure stays, values change	Dates, numeric IDs for testing

Guardrails

Every command gets checked before it runs. If it breaks a rule, it is blocked. If it needs a second pair of eyes, it enters an approval workflow. The model does not decide. Your policies do.

Runbook validation

Pre-built query templates where every parameter is checked against a pattern. Commands stay inside approved shapes. Injection does not get through.

Approval workflows

Risky commands trigger a Slack notification to the right approvers. They see the exact query, not a vague access request. Multi-stage approval for high-risk actions.

Connection segmentation

One connection per risk level: prod-db-readonly, prod-db-write, prod-db-admin. Each with its own rules, its own approvers, its own guardrails.

Outbound allowlisting

Only approved destinations receive outbound traffic. Every request is logged. Nothing leaves your environment without a record of where it went and why.

SUPPORTING COVERAGE

Data subject request readiness

ARTICLES 15 THROUGH 20: ACCESS, RECTIFICATION, ERASURE, AND PORTABILITY

Articles 15 through 20 establish data subject rights: access requests, rectification, erasure, and data portability. Responding to these requests requires knowing where personal data lives and who has touched it.

Hoop.dev provides supporting evidence. Session logs create a queryable record of every data access event, supporting the data mapping and lineage work that underpins effective DSR response.

1 Locate

Query audit logs to identify which databases and tables contain a data subject's records, based on historical access patterns.

2 Attribute

Determine which engineers accessed a data subject's records, when, and for what purpose. Full identity chain from SSO through query execution.

3 Verify

Confirm deletion or rectification by querying post-action access logs. Prove that subsequent access no longer returns the subject's data.

> **Why 'Supporting' and not 'Direct'**
>
> Hoop.dev generates the access evidence layer. It does not manage data inventories, execute deletion workflows, or generate DSR response documents. Those capabilities require purpose-built data governance platforms. Hoop.dev provides the infrastructure access records those platforms need.

RISK EXPOSURE

Quantifying GDPR penalty risk from uncontrolled access

GDPR fines apply for unauthorized access to personal data, failure to implement appropriate technical measures, or inadequate audit trails. The maximum penalty is the greater of €20 million or 4% of global annual turnover.

€20M
MAXIMUM FIXED PENALTY

4%
OF GLOBAL ANNUAL TURNOVER

The highest-risk vector

The highest-risk vector is almost always internal: engineers with direct database access. These are authorized users accessing authorized systems. The risk is not malicious intent. The risk is uncontrolled exposure: no masking, no field-level audit, no evidence that access was proportional.

DPA investigations focus on whether the organization implemented appropriate technical measures. "We have a policy" is not evidence. "We have a gateway that enforces masking on every session and logs every query" is evidence.

> **Risk reduction model**
>
> Hoop.dev reduces GDPR infrastructure risk across three dimensions: it eliminates unmasked PII exposure in query results, generates continuous audit evidence for DPA requests, and removes credential storage as an attack surface. Each dimension maps directly to a GDPR article obligation.

CONTROL GLOSSARY

How gateway controls map to GDPR articles

Each control operates at the protocol layer and runs automatically on every session. No configuration per connection. No rules to maintain.

MASK

AI Data Masking

Redacts 150+ PII types in real time at the protocol layer. Data is filtered before any human or NHI sees it. Generates primary evidence for Article 25 (protection by default) and Article 32 (pseudonymization).

REVIEW

JIT Access Reviews

Time-bounded sessions with approval gates. No standing credentials. Tokens scoped to a task and revoked when done. Generates primary evidence for Article 5(1)(f) (least privilege and access limitation).

KILL

Firewall Policies and Command Interception

Blocks dangerous query patterns. Controls outbound traffic. Risk profiles change per session with no redeployment. Generates primary evidence for Article 32 and Article 5(1)(f).

AUDIT

Immutable Session Recording

Full request/response traceability for every human and NHI session. Every command, every identity, every response recorded. Generates primary evidence for Article 5(2) and Article 30. Supports Articles 15 through 20.

> **Certification scope**
>
> Hoop.dev holds SOC 2 Type II certification and is GDPR compliant. GDPR article references in this guide describe controls for which hoop.dev generates direct or supporting evidence as part of normal gateway operation.

Every database query is a processing activity. Enforce GDPR at the access layer.

Deploy the gateway once. Cover five articles continuously. Hoop.dev enforces Article 25 by default, generates Article 30 evidence automatically, and satisfies Article 32 without configuration.

Documentation:	**hoop.dev/docs**
GitHub:	**github.com/hoophq/hoop**
License:	**MIT**
Contact:	**sales@hoop.dev**