PERCONA

DATABASE SBOM FOR TECHNOLOGICAL SOVEREIGNTY

Verifying Your Data Foundation

A Software Bill of Materials (SBOM) is the "nutrition label" for your database. For an organization seeking sovereignty, the SBOM must prove that the database engine is not a proprietary "black box" controlled by a single vendor.

1. Engine Origin and Upstream Verification

The Goal: Confirm the database is built from verifiable open source code without proprietary "value-add" forks that break portability.

- **Upstream Engine:** Identify the base engine (e.g., MySQL 8.4, PostgreSQL 17).
- **Distribution Provider:** Identify the entity responsible for the build (e.g., Percona, Community, or Cloud Vendor).
- **Fork Analysis:** Flag any proprietary modifications to the storage engine, query optimizer, or replication protocol that do not exist in the upstream codebase.
- **Percona Advantage:** Percona Distributions provide the same "upstream-plus" benefits (enhanced security, performance, and monitoring) across all environments. Because the binaries are identical, you can move workloads between standard Linux servers and Kubernetes clusters without changing a single line of application code.

2. Binary and Build Provenance

The Goal: Ensure the database binaries are signed, auditable, and independent of a specific cloud provider's hardware or hypervisor.

- **Build Environment:** Was the binary compiled in a transparent environment, or is it a proprietary "image" provided by a cloud vendor?
- **Signing and Integrity:** Are the packages signed with a verifiable GPG key to prevent supply chain tampering?
- **Binary Consistency:** Do you have the freedom to use operators, use servers, or use both while maintaining the exact same binary version and patch level?
- **Percona Advantage:** We provide signed, public repositories for all major Linux distributions and container registries. This ensures that your security team can audit the same software layer regardless of your chosen infrastructure.

3. Extension and Library Inventory

The Goal: Audit all "helper" libraries and extensions to ensure they do not create a secondary layer of lock-in.

- **Shared Libraries:** List all linked libraries (OpenSSL, glibc, etc.) and ensure they are standard versions.
- **Proprietary Extensions:** Identify any vendor-specific extensions (e.g., cloud-specific authentication plugins) that would fail to run on a standard server.
- **Audit Capability:** Can your SIEM or security scanner peer into the database environment to verify these versions, or is the environment "locked down" by the vendor?

4. Operational Tooling and Automation Artifacts

The Goal: Treat your database automation as part of your software stack.

- **Orchestration Layer:** Is the logic for backups and scaling trapped in a proprietary cloud API?
- **Automation Portability:** Does your SBOM include the scripts, Helm charts, or Ansible playbooks required to recreate the environment from scratch on independent hardware?
- **Percona Advantage:** By using Percona's portable automation tools, your "Operational SBOM" becomes hardware-independent. Whether you are running on bare metal or in a cloud-native K8s environment, your operational procedures remain identical.

Sovereignty Scorecard: Is Your Stack Portable?

Component	Level 1: Dependent	Level 4: Sovereign (Percona)
Engine	Proprietary Cloud Fork	**Percona Distribution (Upstream Compatible)**
Deployment	Vendor-only Interface	**Standard Servers, K8s, or Hybrid**
Binaries	Opaque / Vendor-managed	**Signed, Auditable, and Universal**
Automation	Proprietary APIs	**Portable Operators and Playbooks**

PERCONA®

Visit Percona's Sovereignty Resource Center or contact us.

CONTACT US