PG_TDE
Toulouse - PG user group



Dimitri Vanoverbeke - Solutions Engineer

PERCONA

© 2026 Percona

Who are we?

Percona Quick overview

TRUE OPEN SOURCE

We believe that open source software—and the community around it—offers the best environment for sustainable growth and genuine innovation without vendor lock-in. You own access and availability of your data.

MULTI-DATABASE READY

Multiple databases for thousands of customers of all sizes and scale? All in a day's work. We handle multi-database complexities, so you don't have to.

INNOVATE FASTER

We enable you to scale with speed, develop with velocity, and exceed customer expectations without worrying if your database infrastructure will be able to keep up.

DEPLOY ANYWHERE

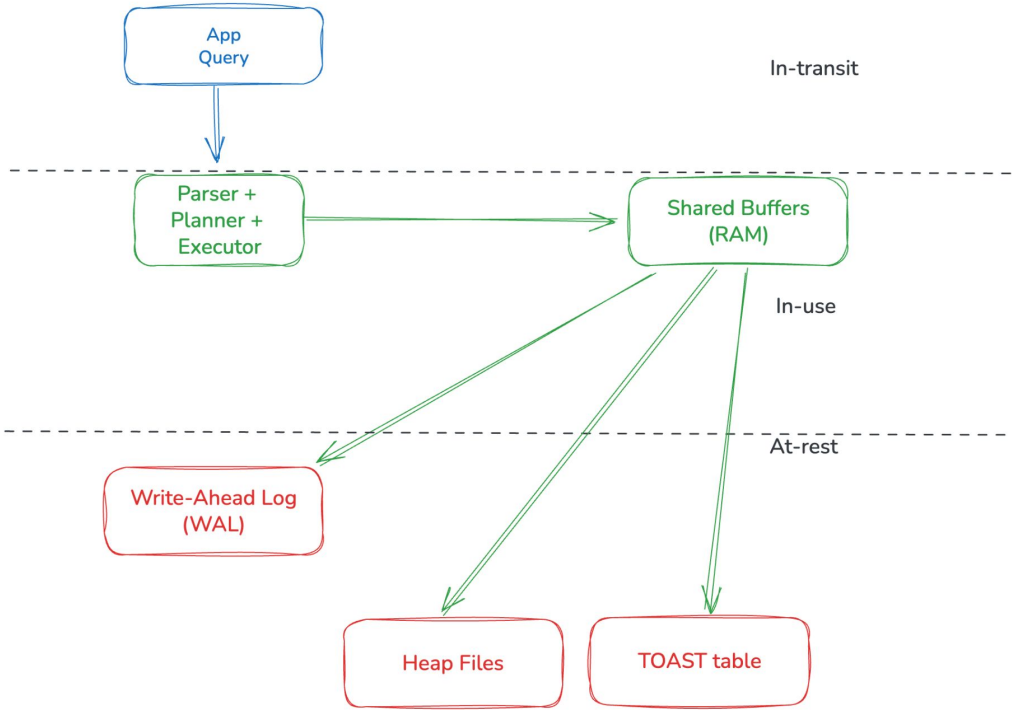
Run your database wherever you want, whenever you want. We have the expertise and open source software to help you manage your database environment on any infrastructure. (any cloud, any database)

PERCONA®

© 2026 Percona

Normal Behaviour

How does Postgres store data on disk?

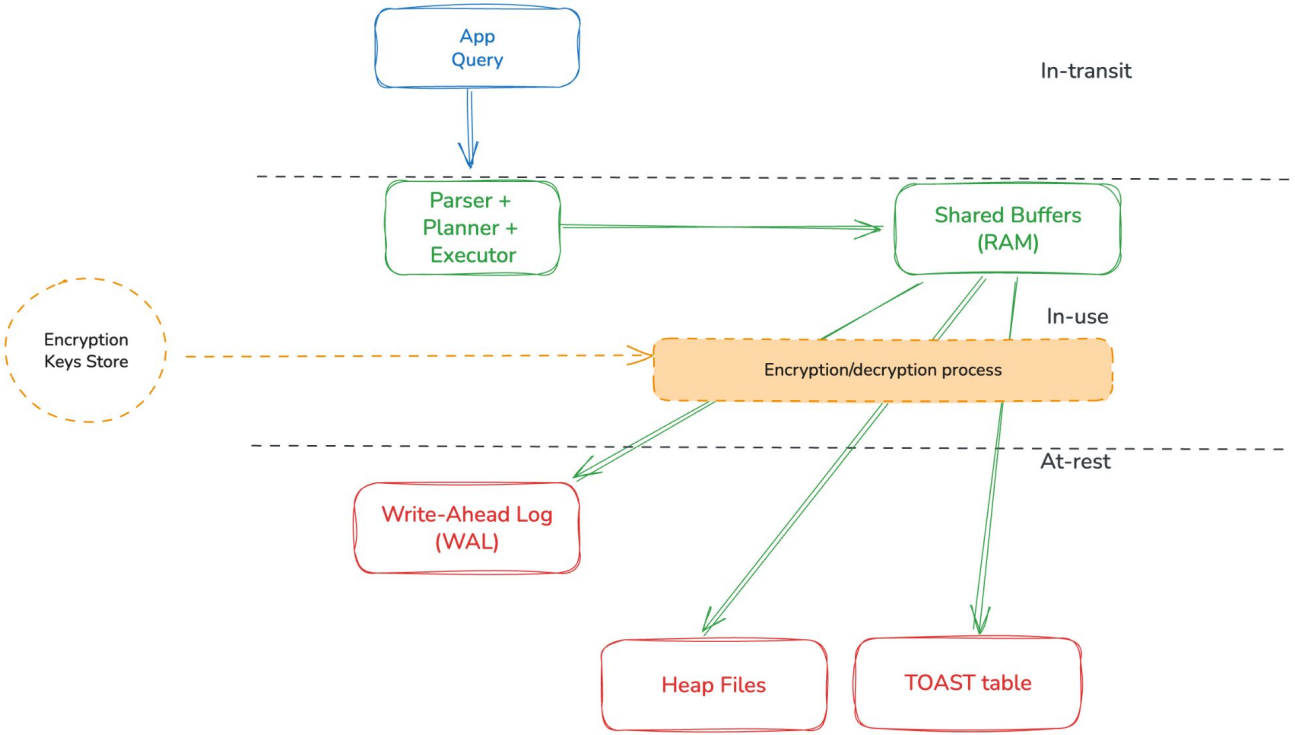


PERCONA

© 2026 Percona

Conceptual

What is TDE



PERCONA

© 2026 Percona

But… Postgres has encryption, no?

- pgcrypto
- pgsodium

Why don’t you just encrypt your storage?

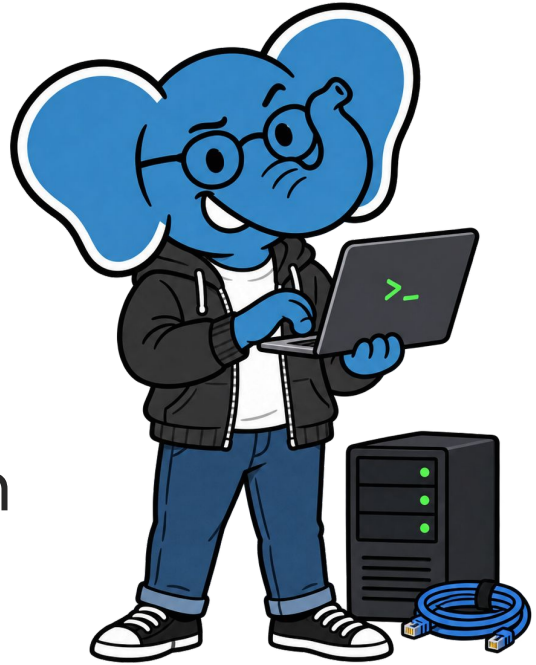
PERCONA®

© 2026 Percona

Why?

Technical reasons

1. Flexibility and granularity
2. Transparency to the application
3. Data files exposure != breach
4. Independence from storage
5. Defence in depth



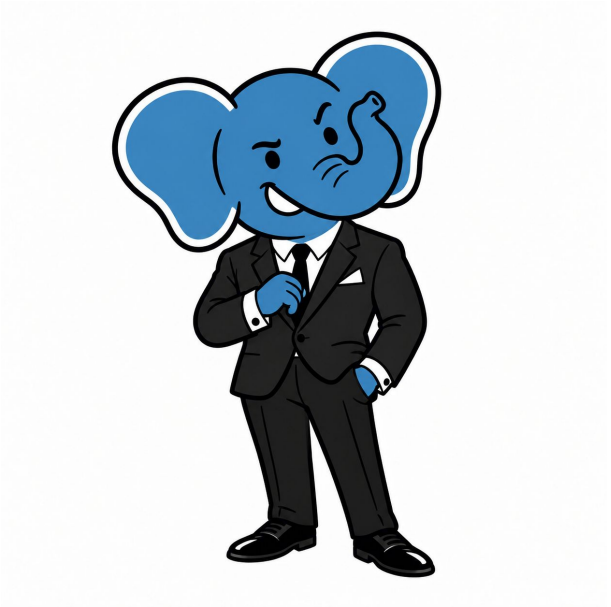
PERCONA®

© 2026 Percona

Non-technical (organizational) reasons

1. Simplified processes
2. Data breach consequences
3. Encryption enforcement

4. Compliance



PERCONA®

© 2026 Percona

Where does the need come from?



PCI DSS

HIPAA

GDPR

LGPD

CCPA

SOC 2

ISO 27001

GLBA

PERCONA

© 2026 Percona

Common functional requirements

1. **Encryption at Rest**
   TDE provides data at rest encryption for storing database information to disk securely.

2. **Key Rotation**
   TDE employs key rotation of master keys to meet compliance rotation requirements as well as incident response if key material is accessed

3. **Access Controls**
   TDE employs the principle of least privilege. Key material is not available to non privileged postgres users. Only a postgres administrator can configure key usage.

4. **Encryption in Transit**
   Separately, postgres supports TLS with OpenSSL for protecting data in transit.

5. **Documented Key Usage**
   TDE encryption is fully documented to expedite cryptographic review for compliance. AES CTR, AES CBC, AES GCM used with appropriate random IVs everywhere and randomized keys for table keys. Master keys are stored in a secure key store separate from TDE.

PERCONA®

© 2026 Percona

Requirement 3.5: Primary account number (PAN) is secured wherever it is stored

Requirement 3.6: Cryptographic keys used to protect stored account data are secured

Requirement 3.7: Where cryptography is used to protect stored account data, key management processes and procedures covering all aspects of the key lifecycle are defined and implemented

© 2026 Percona

Control A.8.24 – Cryptography: Information shall be protected through the use of cryptographic controls

Control A.8.24.1 – Cryptographic Controls Policy: Policy on the use of cryptographic controls for protection of information shall be developed and implemented

Control A.8.24.2 – Key Management: Policy and procedures for the key management shall be developed and implemented

ISO 27001/27002

PERCONA®

© 2026 Percona

CC6.1 - Implement logical access security software, infrastructure, and architectures to protect information assets from security events

CC6.2 - Register and authorize new users and remove system credentials when access is no longer authorized

CC6.3 - Authorize, modify, or remove access based on roles, responsibilities, and least privilege principles

CC6.6 - Implement logical access security measures to protect against threats from outside system boundaries

CC6.7 - Restrict transmission, movement, and removal of information and protect during transit/storage

SOC 2 Trust Services Criteria - Security Category

PERCONA

© 2026 Percona

USE CASE 01

Cardholder data under PCI-DSS

SCENARIO

A payments schema holds account data in a handful of tables inside a much larger database.

Auditors want stored account data unreadable and key custody separated from the database.

WHAT PG_TDE GIVES YOU

Encrypt only the in-scope tables — indexes and TOAST follow automatically.

Principal keys held in Vault or a KMIP HSM, rotated on a documented cadence.

Key-management rights delegated to a role separate from the DBA who reads the data.

WATCH FOR

Spill files past work_mem stay plaintext — size work_mem for in-scope queries.

Column names and catalogs are visible; TDE is not tokenisation or masking.

Enable WAL encryption too, or account data lands unencrypted in the archive.

Net: pg_tde closes the stored-data and key-separation controls. Access control, masking and session-level protections remain your job.

PERCONA®

© 2026 Percona

USE CASE 02

GDPR and Data Residency

SCENARIO	WHAT PG_TDE GIVES YOU	WATCH FOR
EU personal data sits in a cluster whose backups, snapshots and replicas cross regions.	KMS in-region: the key never leaves the jurisdiction, so a copied data file is inert.	Catalogs and schema travel in plaintext with every copy.
Legal needs a defensible answer to "who outside the EU can read this file?"	One principal key per database — a single cluster can serve several jurisdictions with separate custody. Revoking KMS access is a fast, auditable way to cut off a copy you no longer control.	KMS availability becomes a residency-boundary dependency

Net: key locality turns "where does the data live" into "where does the key live" — a boundary you can actually enforce and proof.

PERCONA®

© 2026 Percona

USE CASE 03

SOC 2 Audit readiness

SCENARIO

An auditor asks for evidence that confidential data is encrypted at rest and that keys are controlled.

You need repeatable artefacts, not screenshots.

EVIDENCE YOU CAN PRODUCE

Key and provider inventory from the pg_tde listing and key-info functions.

Separation of duties from the grant and revoke key-management functions.

A monitored pg_tde_verify_key check proving the key is reachable and current.

Encryption enforcement settings that block new unencrypted tables.

WATCH FOR

Key-access audit trails come from Vault or the KMIP server, not from pg_tde.

A single unconverted table quietly breaks the control — audit access methods on a schedule.

Restore drills must prove the key path works, not just that files copy.

Net: the control is only as good as the inventory. Automate the access-method audit and the key-reachability check.

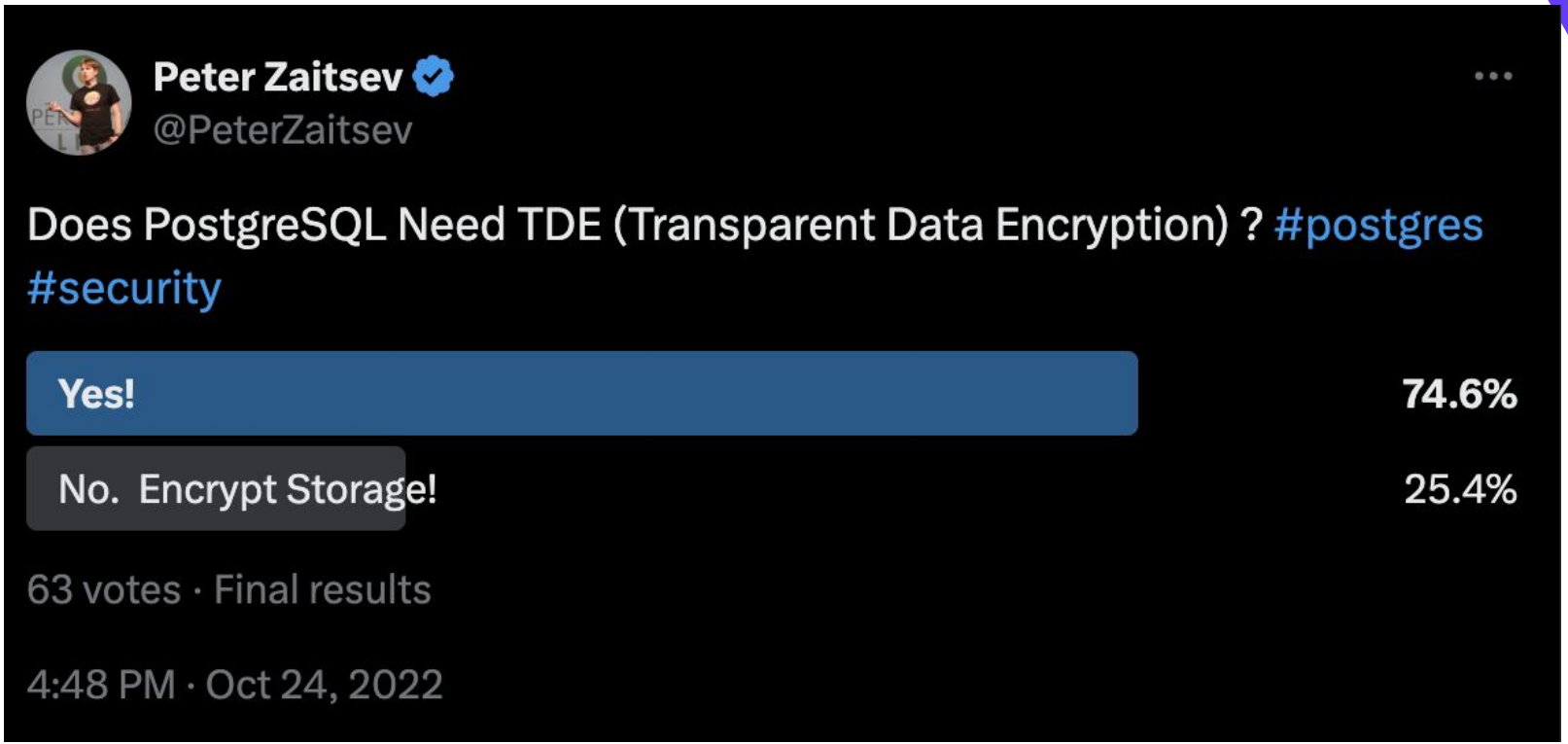
PERCONA®

© 2026 Percona

Rank			DBMS	Database Model	Score		
Oct 2025	Sep 2025	Oct 2024			Oct 2025	Sep 2025	Oct 2024
1.	1.	1.	Oracle	Relational, Multi-model	1212.77	+42.15	-96.67
2.	2.	2.	MySQL	Relational, Multi-model	879.66	-12.11	-143.09
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	715.05	-2.27	-87.04
4.	4.	4.	PostgreSQL	Relational, Multi-model	643.20	-13.97	-8.96
5.	5.	5.	MongoDB	Document, Multi-model	368.01	-12.49	-37.20
6.	6.	↑ 7.	Snowflake	Relational	198.65	+8.46	+58.05
7.	7.	↓ 6.	Redis	Key-value, Multi-model	142.33	-2.84	-7.30
8.	↑ 9.	↑ 14.	Databricks	Multi-model	128.80	+4.74	+43.21
9.	↓ 8.	9.	IBM Db2	Relational, Multi-model	122.37	-1.81	-0.40
10.	10.	↓ 8.	Elasticsearch	Multi-model	116.67	-1.59	-15.18

PERCONA

© 2026 Percona



PERCONA

© 2026 Percona

Timeline allignement

Core changes mean:

- Time to get the patches approved

or

- Carrying patches (~~fork~~)

Meanwhile:

- Time does not pause

and

- Users (customers) have their deadlines
  Compliance policies come into effect



PERCONA

© 2026 Percona

Is TDE new for Postgres?

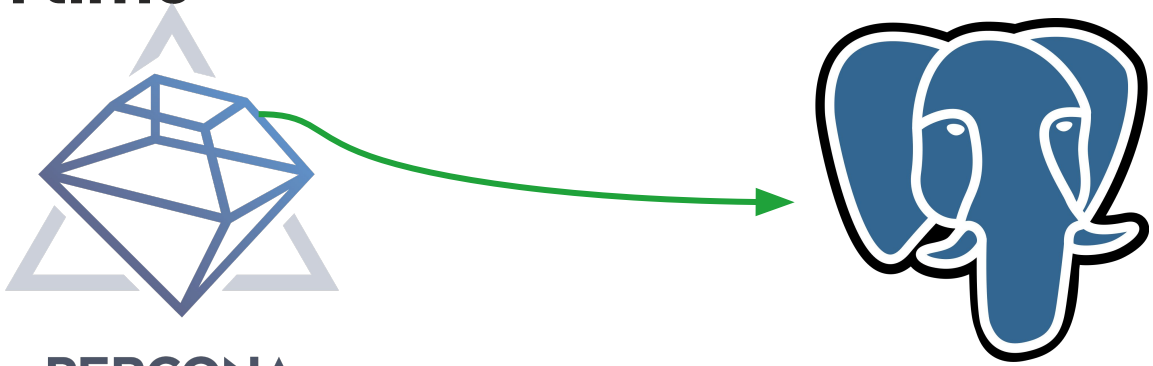


- First patch submitted in 2016
- TDE adopted by proprietary solutions
  - EDB Postgres AI Server
  - EDB Postgres Extended Server
  - Fujitsu Software Enterprise Postgres
  - Crunchy Hardened PostgreSQL (snowflake)
  - Cybertec PostgreSQL Enterprise Edition

PERCONA

© 2026 Percona

Decision time



PERCONA

Distribution for PostgreSQL

Satisfy user needs and provide TDE:

- **right away**
- **fully open source**

Work with PostgreSQL Community to

- **in time provide TDE for all users**

PERCONA

© 2026 Percona

Where?

pg_tde

https://github.com/percona/pg_tde
https://docs.percona.com/postgresql/17/

1. Extension
2. Encrypted access method
3. Open Source - MIT license
4. Developed by Percona
5. Available for Postgres 17+ (specifically Percona Distribution for PostgreSQL)





Percona Distribution for PostgreSQL

Extended PostgreSQL Storage Manager (SMGR) API

```
CREATE TABLE my_table  (id SERIAL, pii_data VARCHAR(32), PRIMARY
KEY(id)) USING tde_heap;
```


PERCONA

© 2026 Percona

What does it Encrypt?

ENCRYPTED

Table data

Indexes

TOAST tables and associated sequences

Temporary tables of encrypted relations

WAL writes, when pg_tde.wal_encrypt is on

NOT ENCRYPTED

System catalogs, statistics and table metadata

Column and table names, comments, schema shape

Query spill files written when work_mem is exceeded

Tables left on the plain heap access method

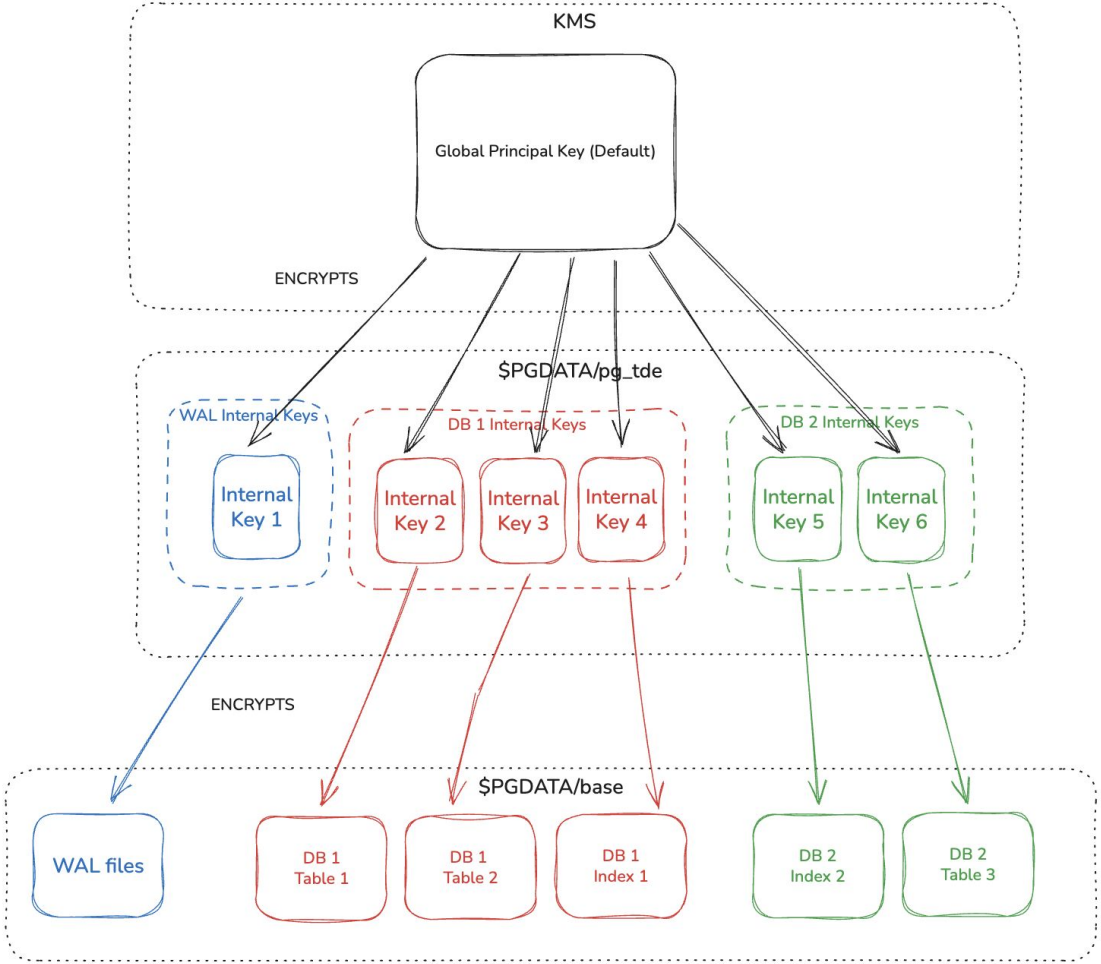
Server logs and anything the application writes elsewhere



*Now also GA!

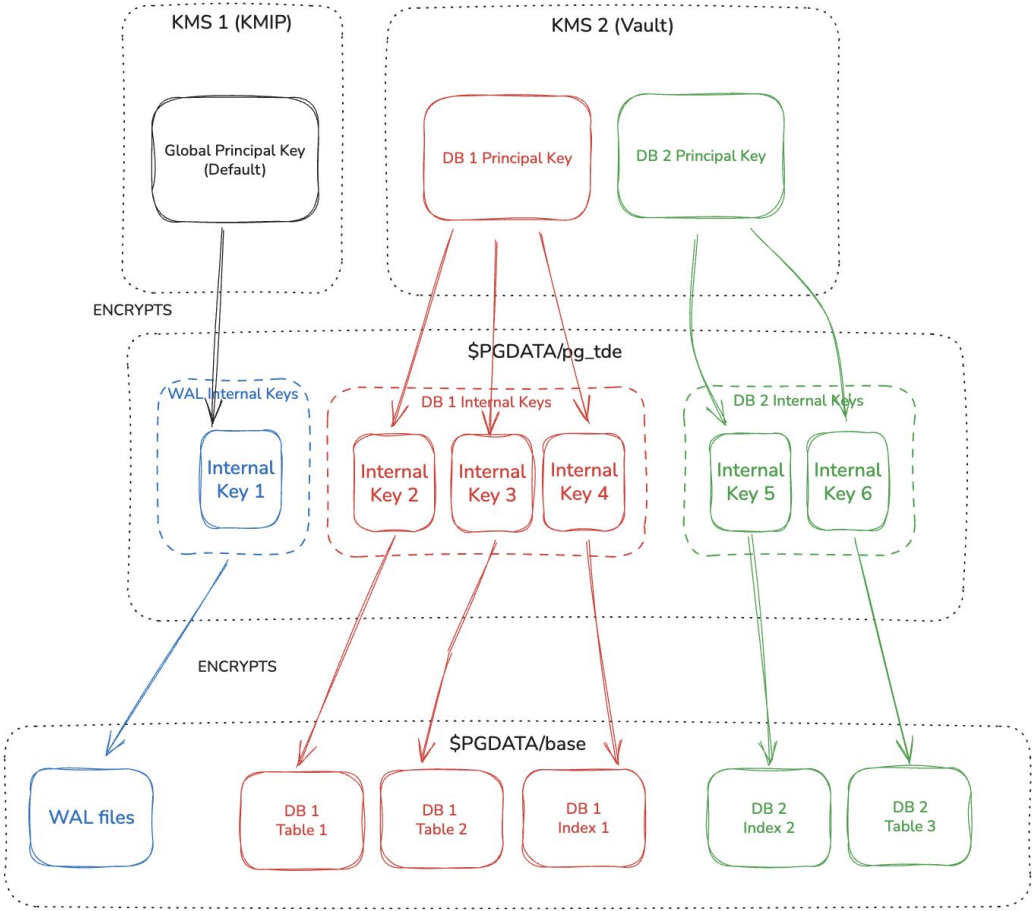
PERCONA®

© 2026 Percona



PERCONA

© 2026 Percona



PERCONA®

© 2026 Percona

Multitenancy

Company A - Database A - Key A
Company B - Database A - Key B
Company B - Database B - Key X

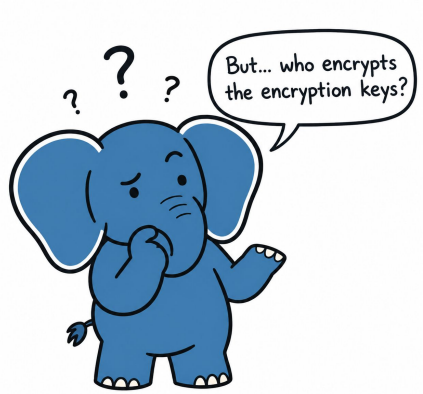
Table level granularity

```
CREATE TABLE customers (
    id BIGINT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    email TEXT
) USING tde_heap;
```

Key-Rotation without re-encrypting data

Key-Management

Thales, Fortanix, Vault, Akeyless and other KMIP solutions



PERCONA

© 2026 Percona

PERFORMANCE

PUBLISHED BENCHMARK RESULTS

PERCONA DOCUMENTATION

<10%

Overhead for most situations. JSONB-heavy operations can degrade further.



INDEPENDENT · PGBENCH, 2025

~20%

WAL encryption on write-heavy runs. Data-page encryption alone was measurable but small.

HOW THE INDEPENDENT TEST WAS RUN

pgbench, read-only and read/write, 1–70 clients, scale factors 5 to 5000 — roughly 62.5 GB at the top end — 1000 transactions per run, best of five rounds, on a 48-core server.

READ THE NUMBERS CAREFULLY

That study used a release candidate while WAL encryption was still beta. Treat ~20% as a ceiling.

Sources · pg_tde documentation · Percona, "The performance cost of transparent data encryption in PostgreSQL" (2026) · A. Scherbaum, independent TDE performance test (2025)

© 2026 Percona

Upcoming

What is next?

https://github.com/percona/pg_tde
https://docs.percona.com/postgresql/17/

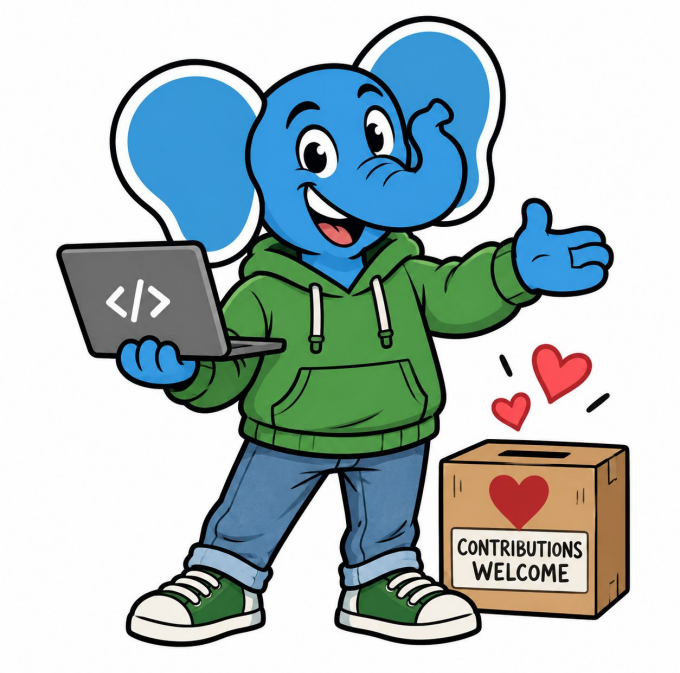
1. PostgreSQL 19 support ⟷ Supported in community?
2. CNPG support being investigated
3. Amazon KMS support

PERCONA®

© 2026 Percona

What you can do?

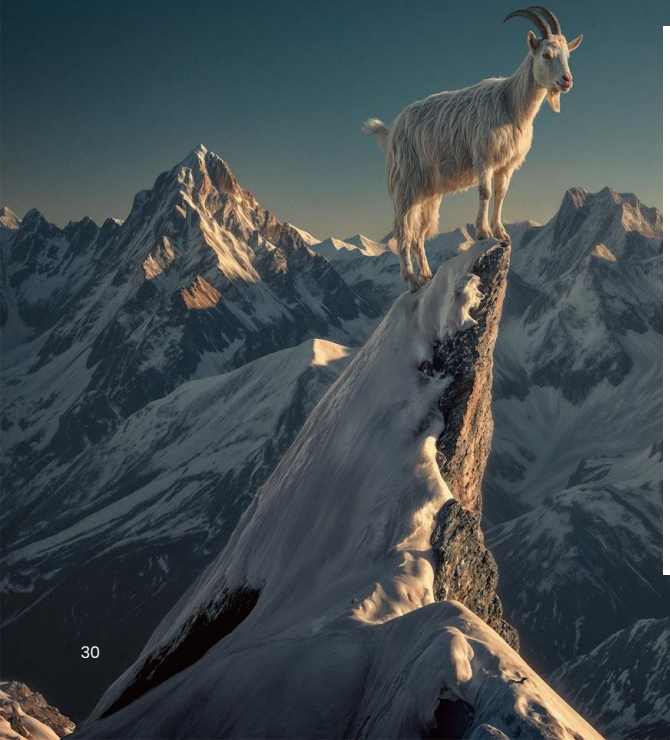
Use it, test it, contribute!
Give us your feedback!



PERCONA

© 2026 Percona

Join us at Percona.Connect PARIS



PERCONA®

© 2026 Percona



PERCONA®

© 2026 Percona



© 2026 Percona