PERCONA

The Elephant Has No Clothes

Jan Wieremjewicz

Who am I

Product manager for PostgreSQL in Percona

“Fun” fact 1: used to be reluctant to prepare public decks due to copyrighted graphics

“Fun” fact 2: still feel awkward about using AI to help with work.

Let’s have fun! In this deck, following this slide, no slide has been left w/o AI generated imagery.

Disclaimer

We're known to be a company that's evangelizing open source

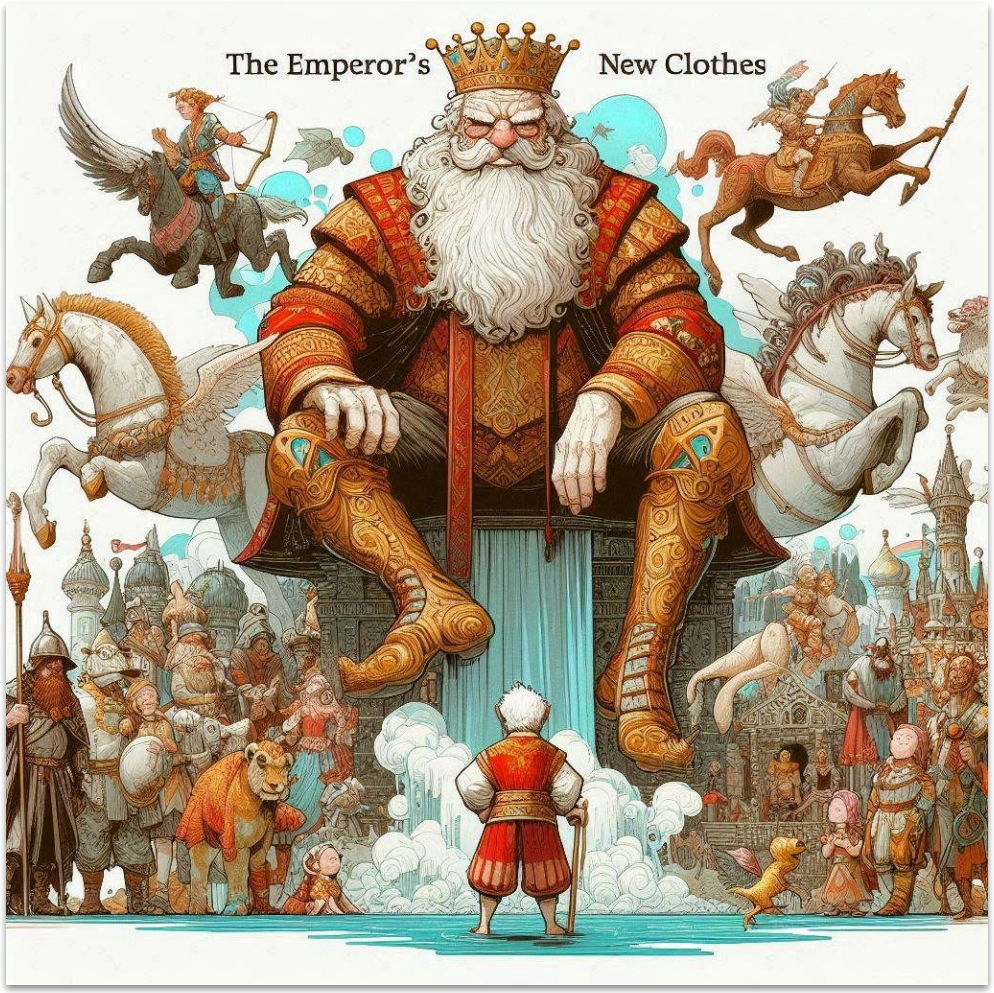
We want to be transparent and open in how we come to the decisions as well

In this presentation I will	Share the story about implementing TDE for PostgreSQL: the why, the what, the how. The PM POV
I will most likely not	Go into deep technical details

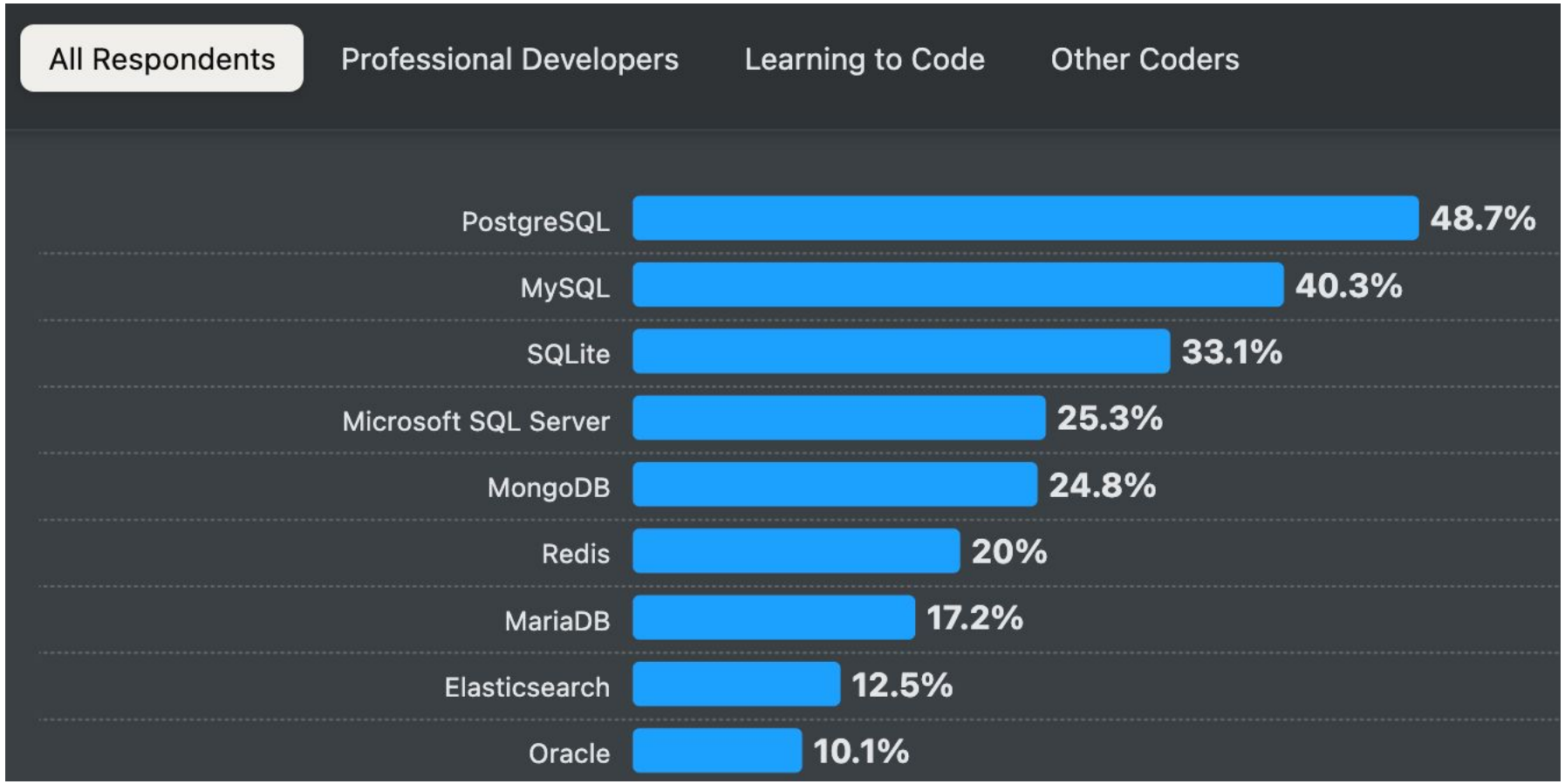
Emperor’s new clothes

Folk tale by Hans Christian Andersen first published in 1837, based on a 1335 story from a medieval Spanish collection of 51 tales with various sources (source)

In “The Emperor’s New Clothes,” the emperor is deceived into believing he’s wearing invisible clothes, and no one dares to tell him the truth until a child points out he’s actually naked.



Who is the Emperor in our story?



Source: Stackoverflow

Let’s face the elephant in the room

Similarly to “Emperor’s New Clothes”, PostgreSQL, despite being a leading database, lacks native data at rest encryption. Like the emperor, it relies on external solutions for security, leaving a critical gap that users often overlook.

Why talk about TDE?

We believe an open world is a better world. Our mission is to enable everyone to innovate freely, by providing the best open source database software, support, and services.

Let’s ask the users, why do they need TDE?

Outsider answers

> ***What do you mean why do we need to get it? It’s already included in PostgreSQL!***

Let’s ask the users, why do they need TDE?

Unfortunately, Nope…

After almost a decade of discussions there still is no open source TDE.

There are **TDE** solutions for PostgreSQL, but no open source ones

Let’s not confuse TDE with:

- pgcrypto
- pgsodium

Let’s ask the users, why do they need TDE?

Newbie answers

> ***What's the point of encrypting data at rest in the application-level with AES/RSA encryptions if I'm already using HTTPS?***

Before we go further

Newbie answers

What's the point of encrypting data at rest in the application-level with AES/RSA encryptions if I'm already using HTTPS?

AI: what is TDE?

Transparent Data Encryption (TDE) is a security technology used to encrypt data at rest, meaning the data stored in database files on disk

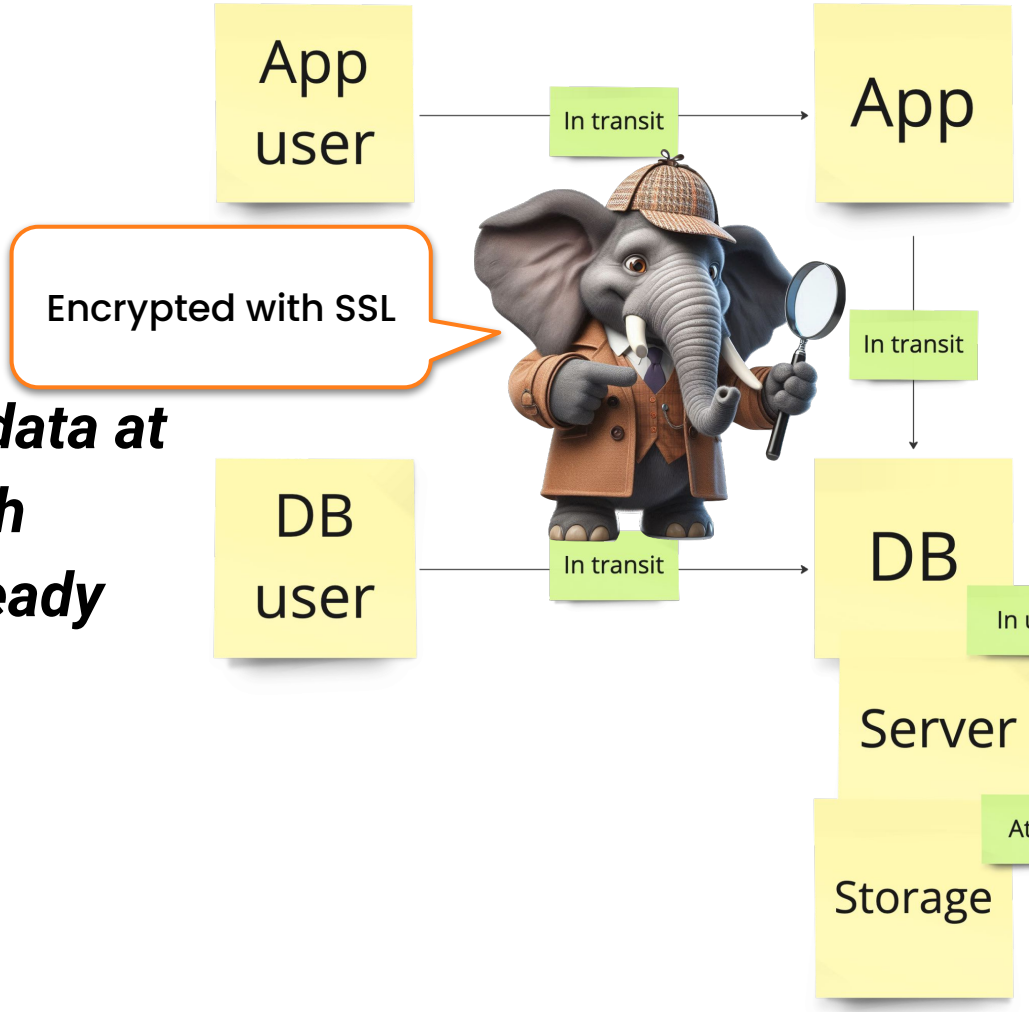
(...)

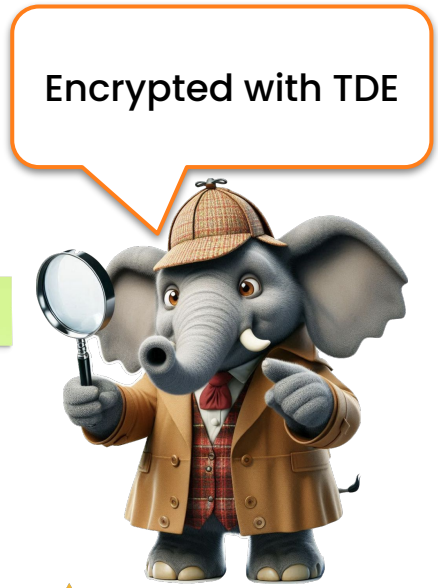
TDE ensures that even if someone gains physical access to the storage media, they can't read the data without the proper decryption keys

Before we go further

Newbie answers

What's the point of encrypting data at rest in the application-level with AES/RSA encryptions if I'm already using HTTPS?

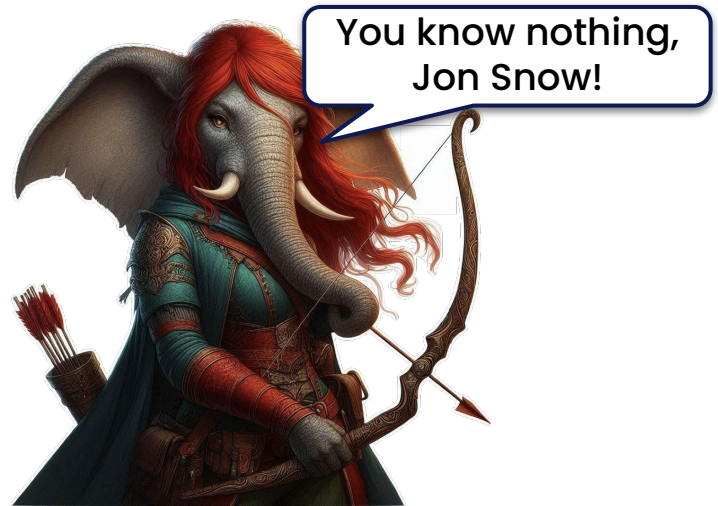


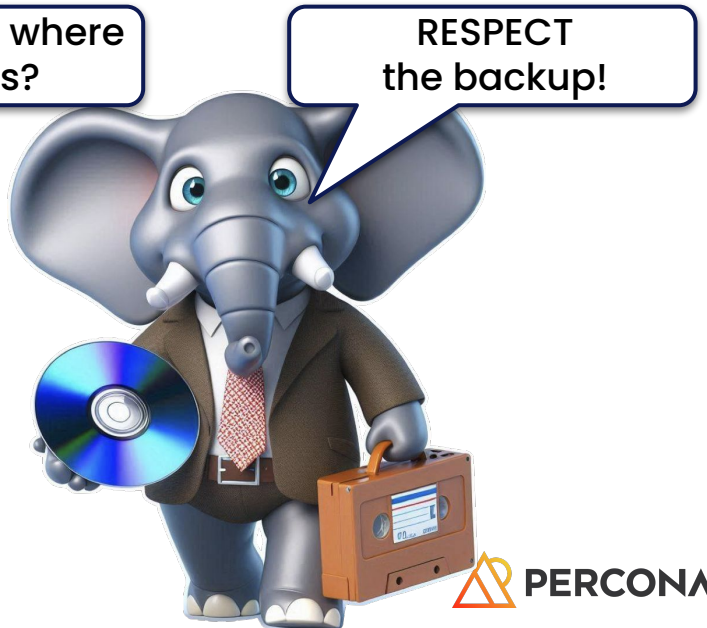


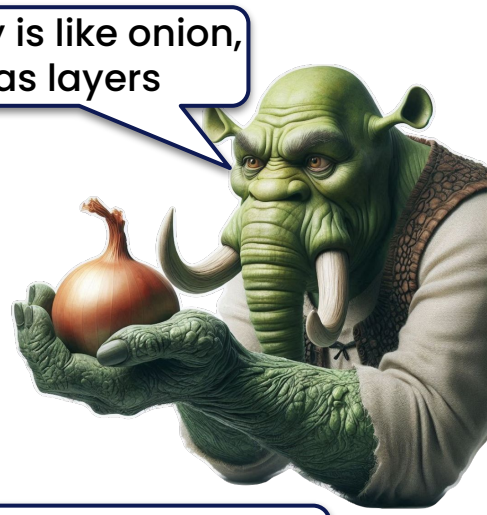
Let’s ask the users, why do they need TDE?

Dev answers

Technical solutions exist







PERCONA

Let’s ask the users, why do they need TDE?

Business answers

policies

compliance

standards

regulations

internal regulations

risk management

security

Let’s ask the users, why do they need TDE?

Business answers

compliance

policies

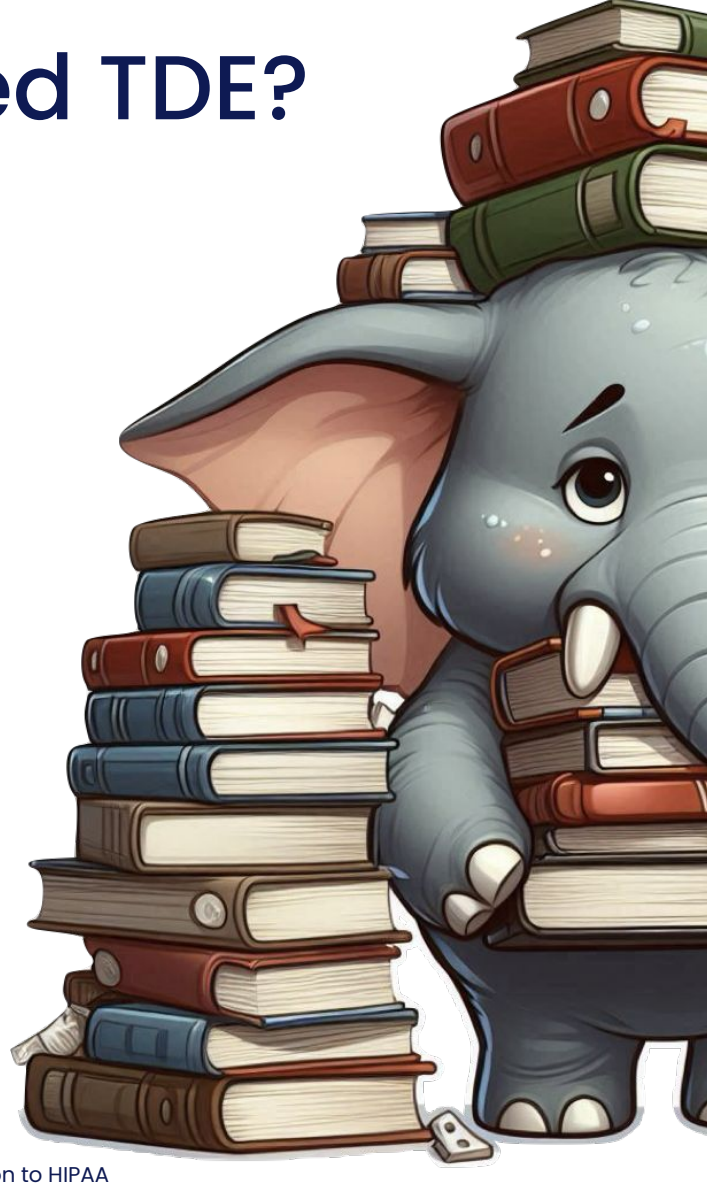
standards

regulations

internal regulations

risk management

security



- PCI DSS
- GLBA
- GDPR
- LGPD
- CCPA
- SOC 2
- HIPAA
  - HITECH - extension to HIPAA
- ISO 27001:2013 Annex A.10

Observation

A lot of policies do not consider encrypted data leaks as a data leak

Reason: it cannot be proven that the data has been successfully accessed

- PCI DSS
- GLBA
- GDPR
- LGPD
- CCPA
- SOC 2
- HIPAA
  - HITECH - extension to HIPAA
- ISO 27001:2013 Annex A.10

If only they encrypted data

2005 - 2009 UCLA Hospitals - unauthorized access

2010 - 2013 Children's Medical Center of Dallas - not encrypted

2011 - Tricare - encryption didn't match with federal standard

2013 Yahoo! - easy to break encryption

2013 - Advocate Health Care - data not encrypted

2013, 2017 New York Medical Center - data not encrypted

2017 - Lifespan Health System - data not encrypted

2020 - COSMOTE Mobile Telecommunications - not encrypted

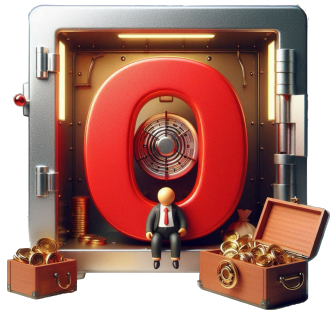
2021 - UK Home Office - data not encrypted

2022 - Tuckers Solicitors - data not encrypted

2024 - National Public Data - allegedly data not encrypted

Trunk load of databases include TDE

	MySQL	Oracle	MS SQL
Encryption Type:	• Block-level encryption	• Block-level encryption	• Block-level encryption
Encryption Downtime	• Performed online	• Performed online	• Performed online
Encryption Algorithm:	• AES-256	• AES-128, AES-192, AES-256, 3DES.	• AES-128, AES-192, AES-256, 3DES
Encryption Level:	• Tablespace	• Tablespace, column	• Database
Key management:	• Oasis KMIP, Oracle Cloud Infrastructure Vault, Hashicorp Vault, AWS KMS	• Oracle Wallet, Oracle Key Vault, and KMIP support.	• Database Master Key, Certificates, integration with Azure Key Vault, AWS KMS, and KMIP support.



What can we learn from them?

Must have

- Lower than cluster granularity
- Integration with external KMS
- Online encryption - no restarts necessary

Should have

- Config of the encryption algorithm
- Range of KMS integrations

Respect the Community

Community assumptions:

- Secure - protecting the data against all intended attack vectors not against all potential attack vectors
- Minimal impact on the rest of PostgreSQL code
- Meets regulatory requirements

Requirement we do not feel is right:

- cluster wide encryption

Source: TDE wiki

Respect the Community

- Minimal impact on the rest of PostgreSQL code

→ Why implement it in the core, when we have an amazing extensibility in PostgreSQL?

Source: TDE wiki

pgcrypto

- Postgres contrib module included in pgdg Postgres
- Provides cryptographic functions for PostgreSQL
- Popular use cases:
  - Encrypting/decrypting data on column level
  - Hashing data
  - Generating and verifying digital signatures
  - Random data generation
  - Key generation
  - Hashed password handling

- Not transparent
- Column level only
- No indexes on encrypted columns
- User managed keys
  - No KMS integration

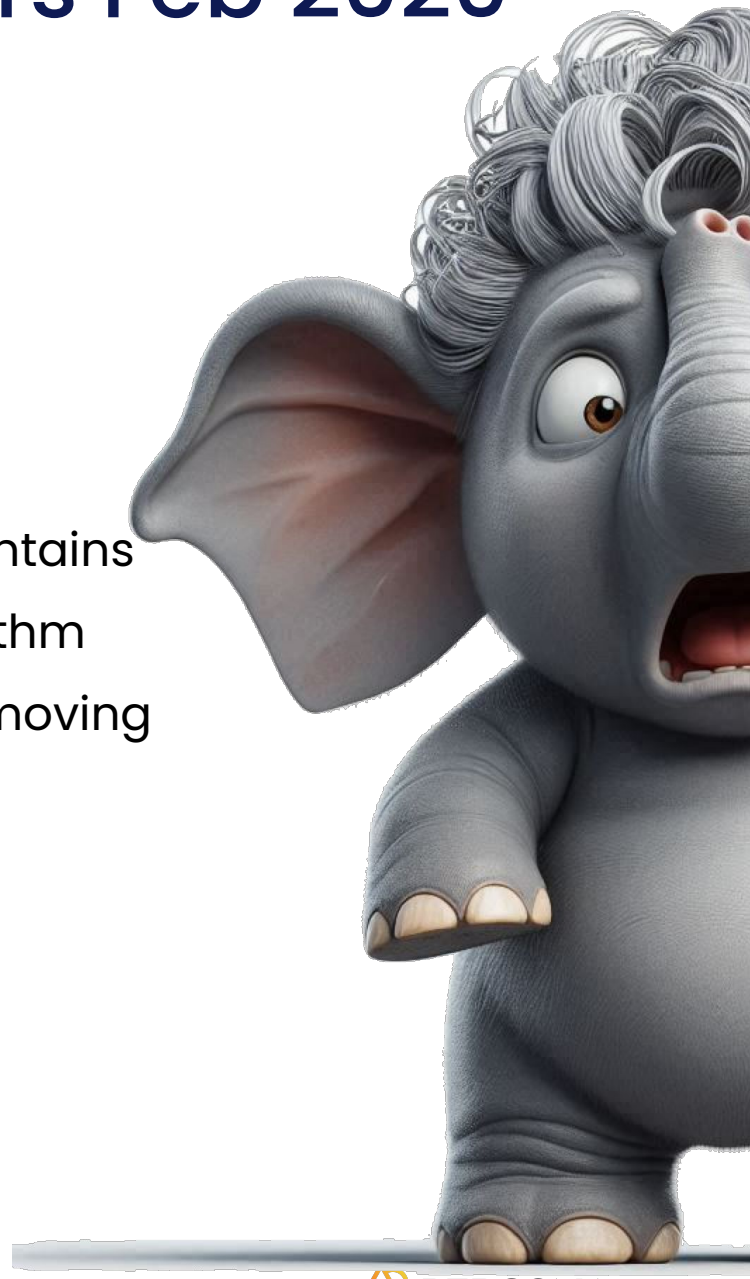
PERCONA

Why not pgcrypto? - pgsql-hackers Feb 2020

From: Andres Freund

> I'd strongly advise against having any new infrastrure depend on
> pgcrypto. Its code quality imo is well below our standards and contains
> serious red flags like very outdated copies of cryptography algorithm
> implementations. I think we should consider deprecating and removing
> it, not expanding its use. It certainly shouldn't be involved in any
> potential disk encryption system at a later stage.

This is an unsolicited public opinion about pgcrypto.



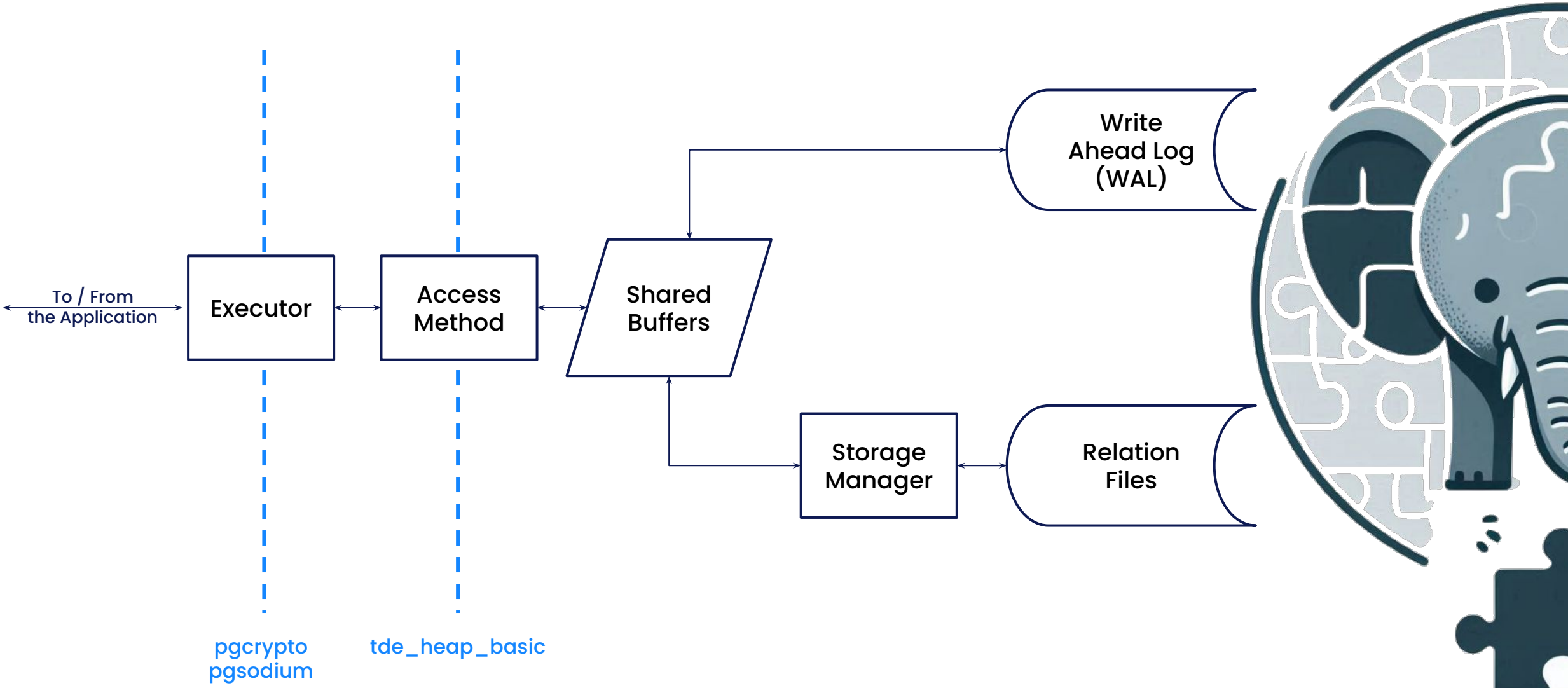
pgsodium

- PostgreSQL extension enabling the usage of a well respected cryptographic algorithms library:
  - Uses libsodium library for high level cryptographic algorithms
  - pgsodium contains no encryption algorithms (lightweight)
  - Wraps the libsodium library 1 to 1
  - Focus on ease of use
- Minimal Server-Key Management
  - Server can preload a libsodium key on server start
- Transparent Column Encryption for one or more columns (not really transparent)

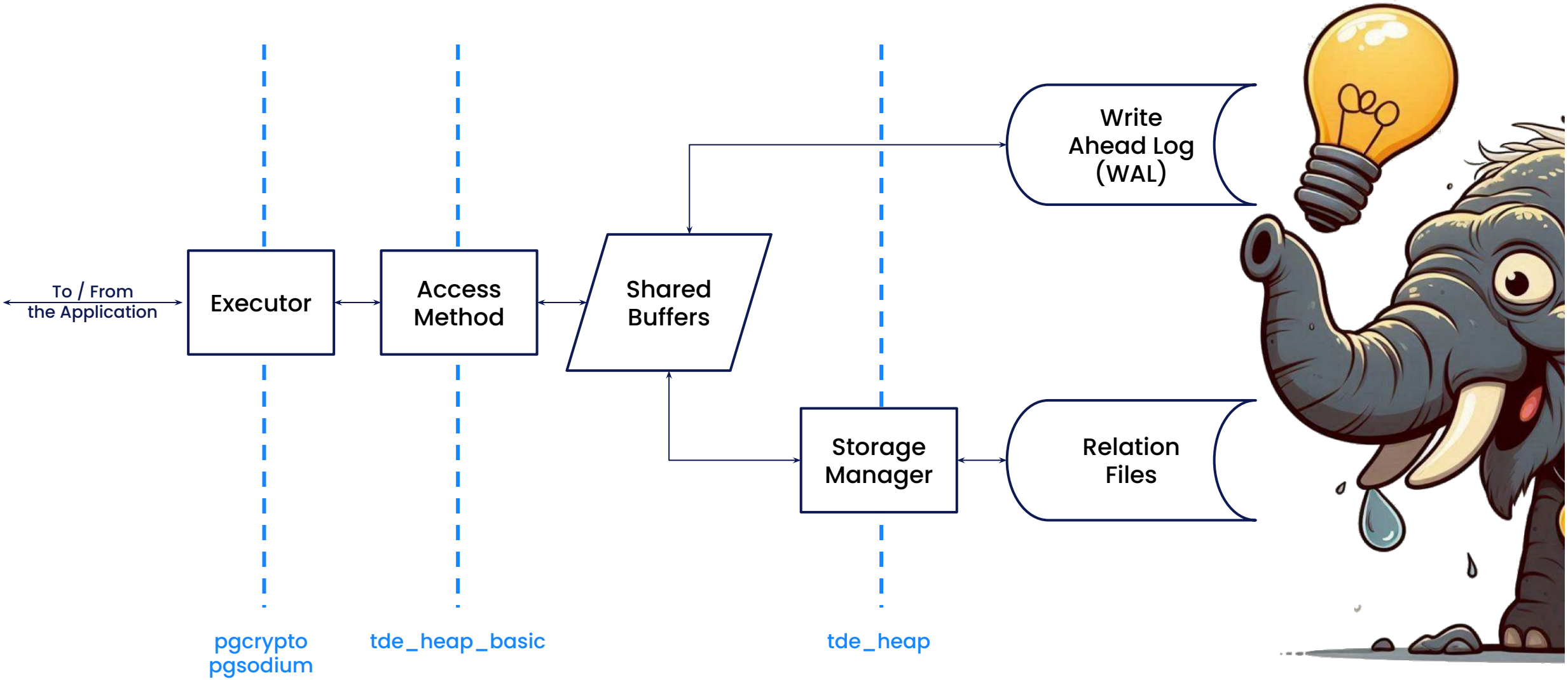
- Not transparent
- Column level only
- No indexes on encrypted columns

PERCONA

So how does TDE work?



So how does TDE work?



That means we need to modify/extend:

Storage Manager (SMGR) API

- ✓ no need to encrypt/decrypt for each shared buffer write/read
- ✓ index data encryption
- ✓ replication support

WAL Read/Write API

- ✓ data encrypted in WAL

That means we need to modify/extend:

Storage Manager (SMGR) API

- ✓ no need to encrypt/decrypt for each shared buffer write/read
- ✓ index data encryption
- ✓ replication support

WAL Read/Write API

- ✓ data encrypted in WAL

Proposed by Neon

Hardening improvements planned as a contribution from Percona

Current implementation very specific to this use case

Encrypting is an easy a change in access method

PostgreSQL Community

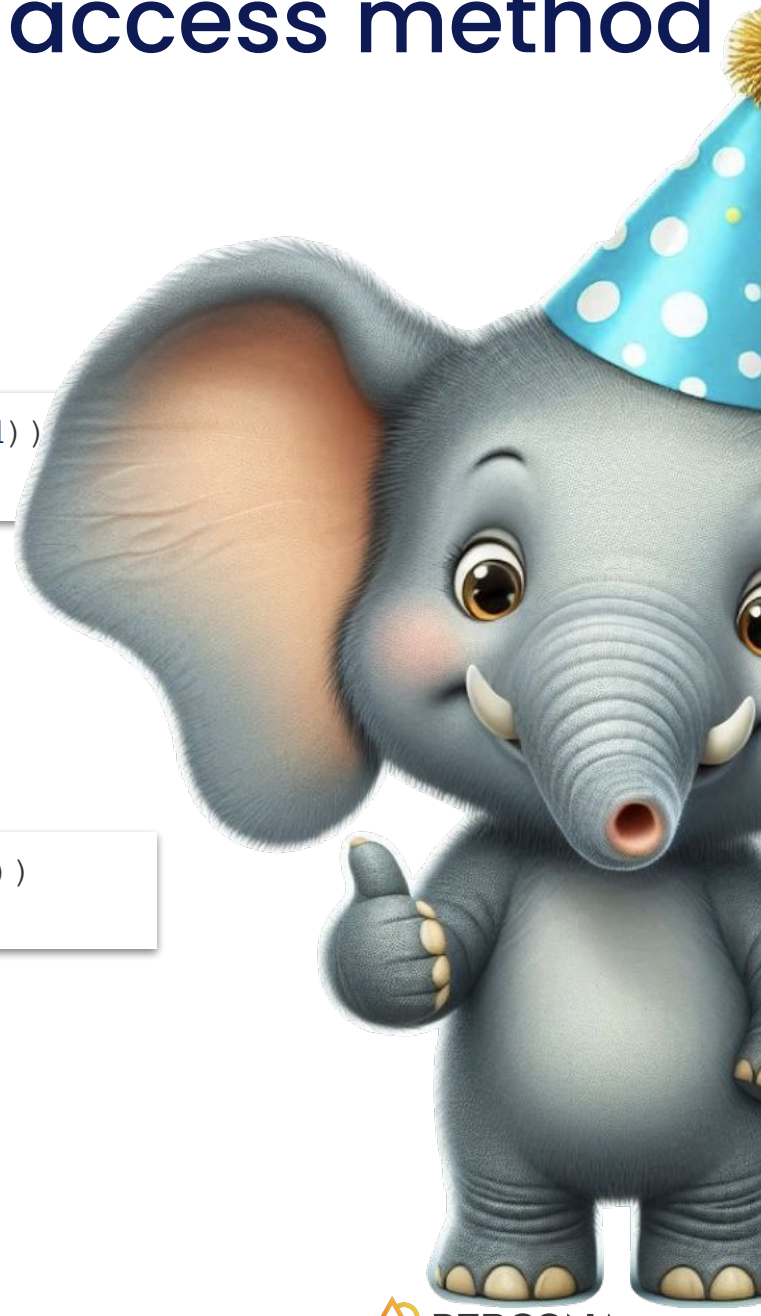
Default PostgreSQL Storage API

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

Percona Distribution for PostgreSQL

Extended PostgreSQL Storage API

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



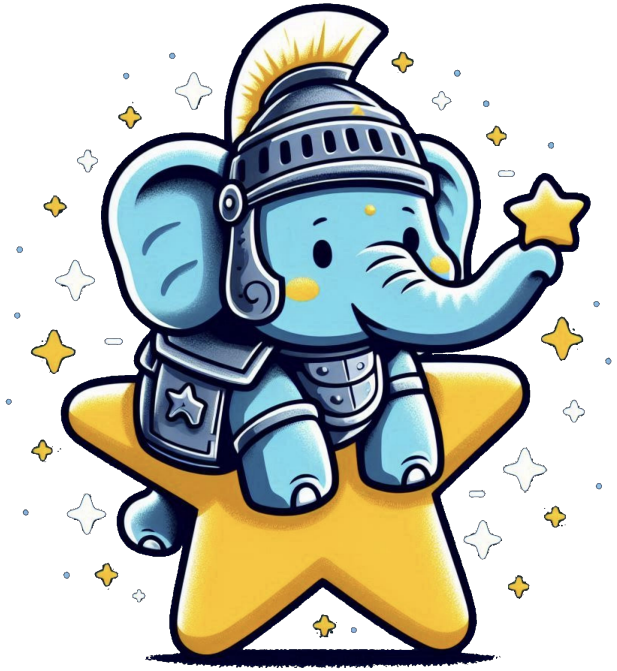
And so… Check it out!

We chose to:

1. Implement changes in the server in the Percona Distribution for PostgreSQL
   - these are still a Tech Preview, impactful only to the users who try the pg_tde build for Percona Server for PostgreSQL
   - available in Percona Distribution for PostgreSQL 17.0.1
2. Provide the pg_tde build for Percona Server for PostgreSQL
3. Contribute the changes to the Community PostgreSQL

https://github.com/percona/pg_tde

And 🌟 the project



https://github.com/percona/pg_tde

What now?

Contribute!

- Code is what we have
- Feedback is what we need:
  - Try it out
  - Let us know how it went!

https://docs.percona.com/postgresql/17/postgresql-server.html

Bow to the king

Similarly to “Emperor’s New Clothes”, PostgreSQL, despite being a leading database, lacks native data at rest encryption. **Like the emperor, it relies on external solutions for security, leaving a critical gap that users often overlook**.

TDE

Beta build for community PG available now!

Contact me about access to Tech Preview

jan.wieremjewicz@percona.com

PERCONA

Databases run better with Percona