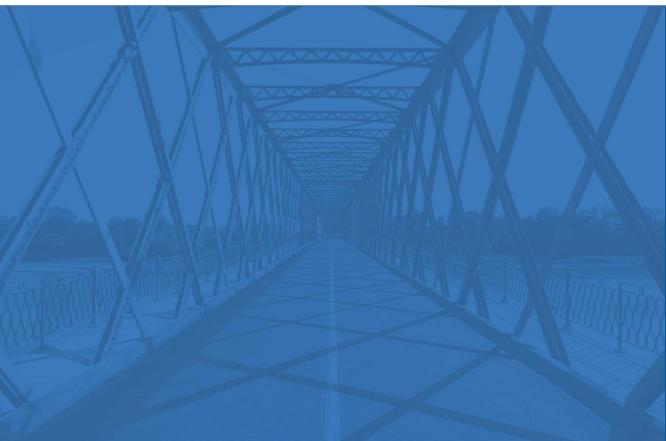




Socle Dalibo Postgres 19

Octobre 2026

Damien Clochard
Product Owner

Vision

- A mesure que PostgreSQL devient la base de données par défaut dans la plupart des organisations, les usages et le nombre d'instances sont en progression constante.
- Le défi des équipes DBA consiste à faire vivre et grandir leur parc d'instances postgres de manière homogène et dynamique.
- Le **Socle Dalibo Postgres** offre un cadre d'exploitation à la fois souple, inter-opérable, fiable et open-source .

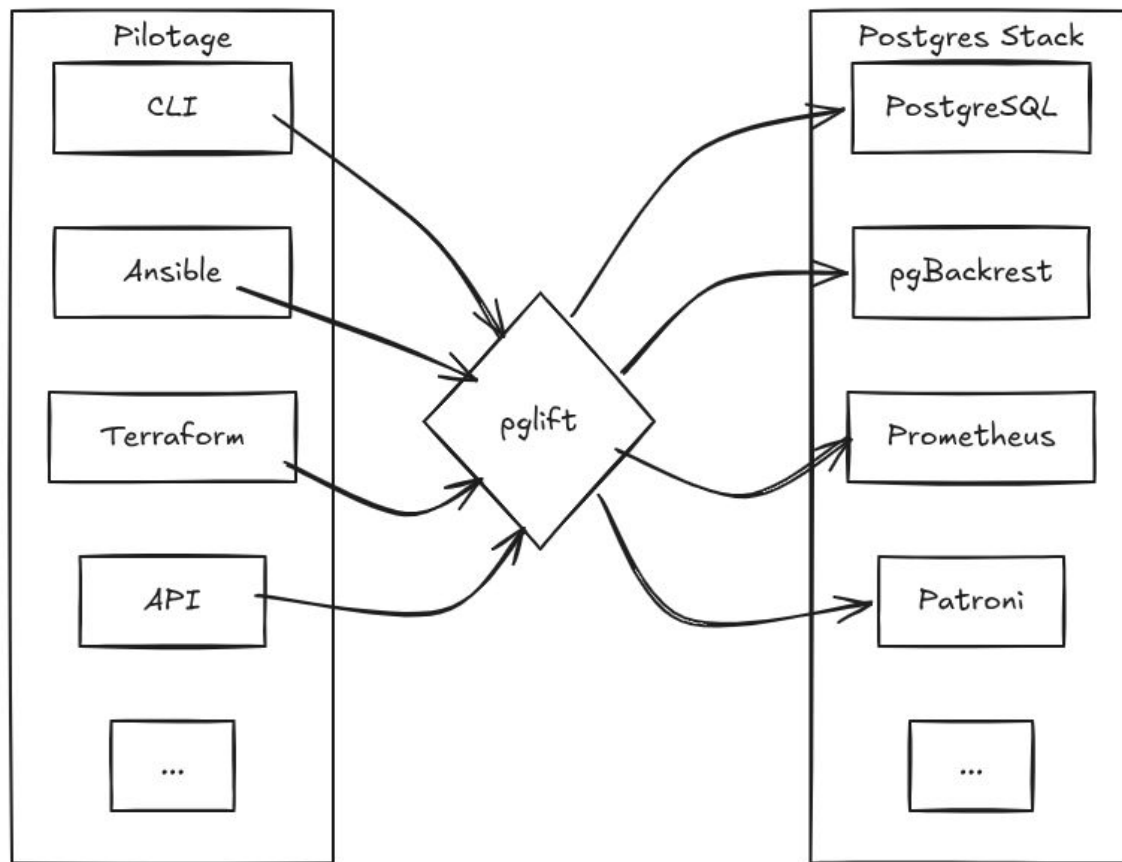
Enjeux

- **Automatisation** : Gérer un parc d'instances en expansion
- **Hybridation** : Mixer les stratégies (Cloud Public/Privé, k8s, ...)
- **Agilité** : S'adapter aux évolutions du marché
- **Indépendance** : Innover sans être lié à un opérateur / éditeur
- **Démocratisation** : Rendre Postgres plus accessible, en particulier pour les développeurs.

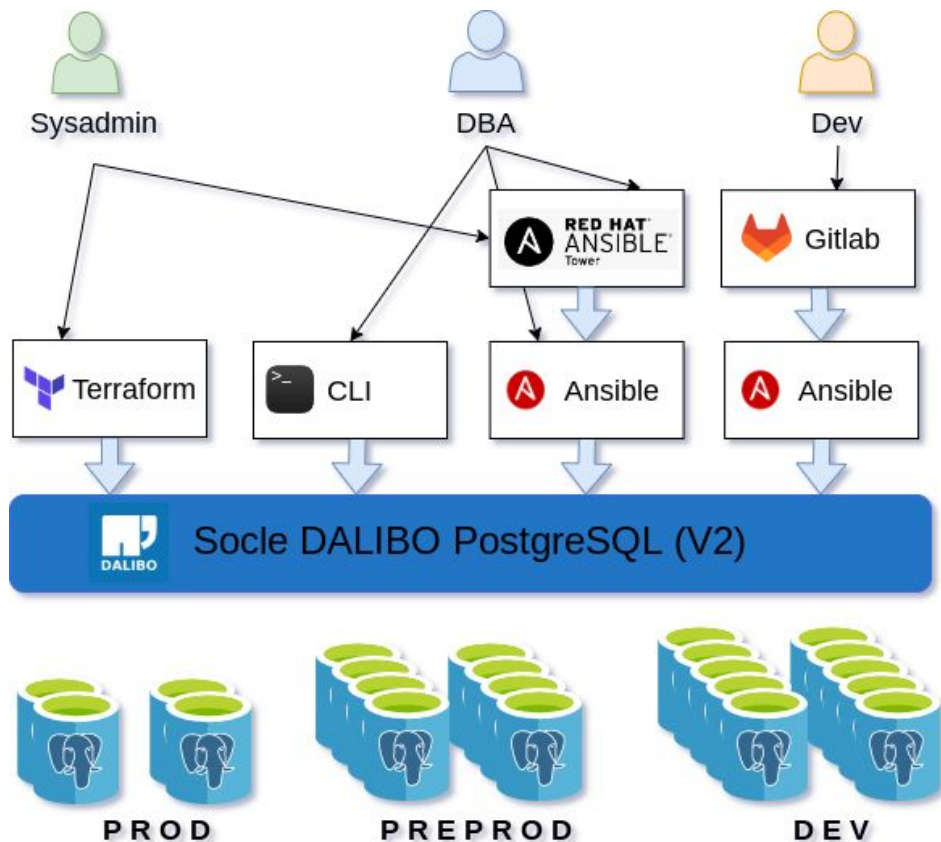
Socle Dalibo Postgres

- 20 ans d'expertise « encodés » dans un socle
- Offre qui s'adresse exclusivement aux grands comptes
- Conçu pour être customisé et adaptable
- La sécurité est un sujet central et évolution constante
- Un panel d'outils 100 % Open-source

pglift ?



Exemples d'Intégration



Socle Dalibo Postgres

3 collections de roles Ansible pour l'installation

1 collection de modules Ansible pour l'exploitation

1 outil en ligne de commande (pglift)

RHEL 8, 9 & 10 / Debian 13 / Ubuntu 26.04

100% Open Source

Socle Dalibo Postgres

Support Niveau 4 et 24/7 sur toute la stack :

PostgreSQL + 20 outils satellites

Workshops et Formations

Manuels et documentation adaptés

1 DBA intégrateur référent

Points Clés

Des modules Ansible qui respectent l'idempotence

Entièrement pilotable via Red Hat Automation Platform

un Module Terraform / OpenTofu

une API Rest

Interfaces possibles avec d'autres technos Infra-as-Code

Un outil CLI pour passer en mode manuel à tout moment

une stack PostgreSQL complète

Observabilité: Intégration de Prometheus

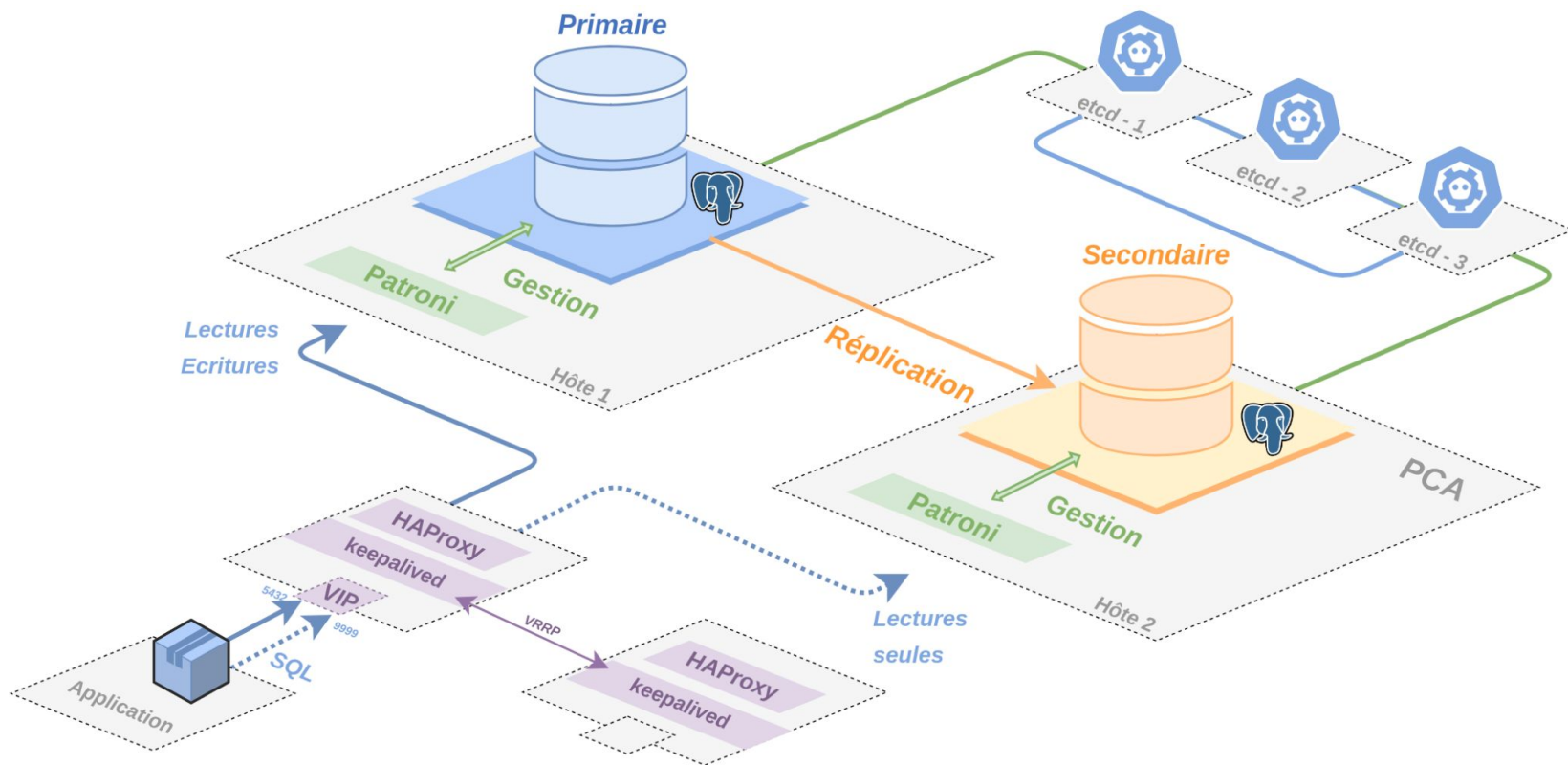
Sauvegardes: locales ou centralisées

Haute-Disponibilité : Patroni & ETCD

Performance : PoWA + pgbadger

Conformité : PostgreSQL Anonymizer / pgAudit

une stack PostgreSQL complète



Cycle de Vie

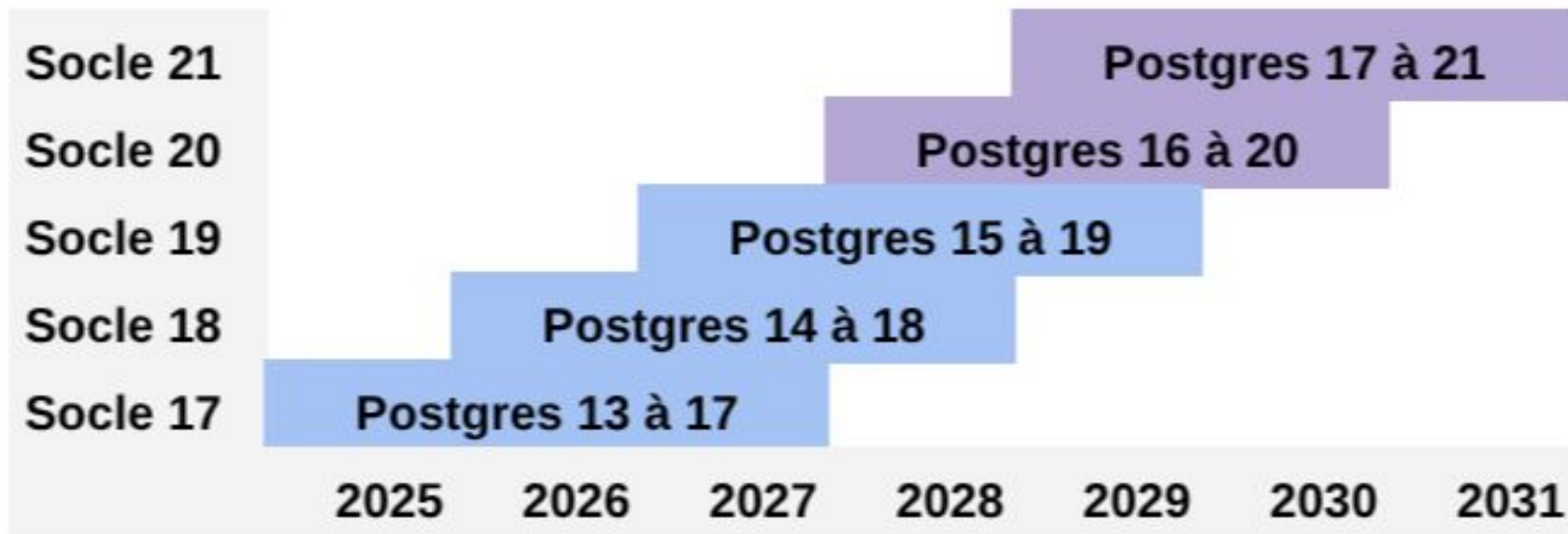
Socle Dalibo PostgreSQL 18

- publiée en janvier 2026
- supporté jusqu'en janvier 2027
- PostgreSQL 14 à 18

Socle Dalibo PostgreSQL 19

- prévue pour janvier 2027
- supporté jusqu'en janvier 2028
- PostgreSQL 15 à 19

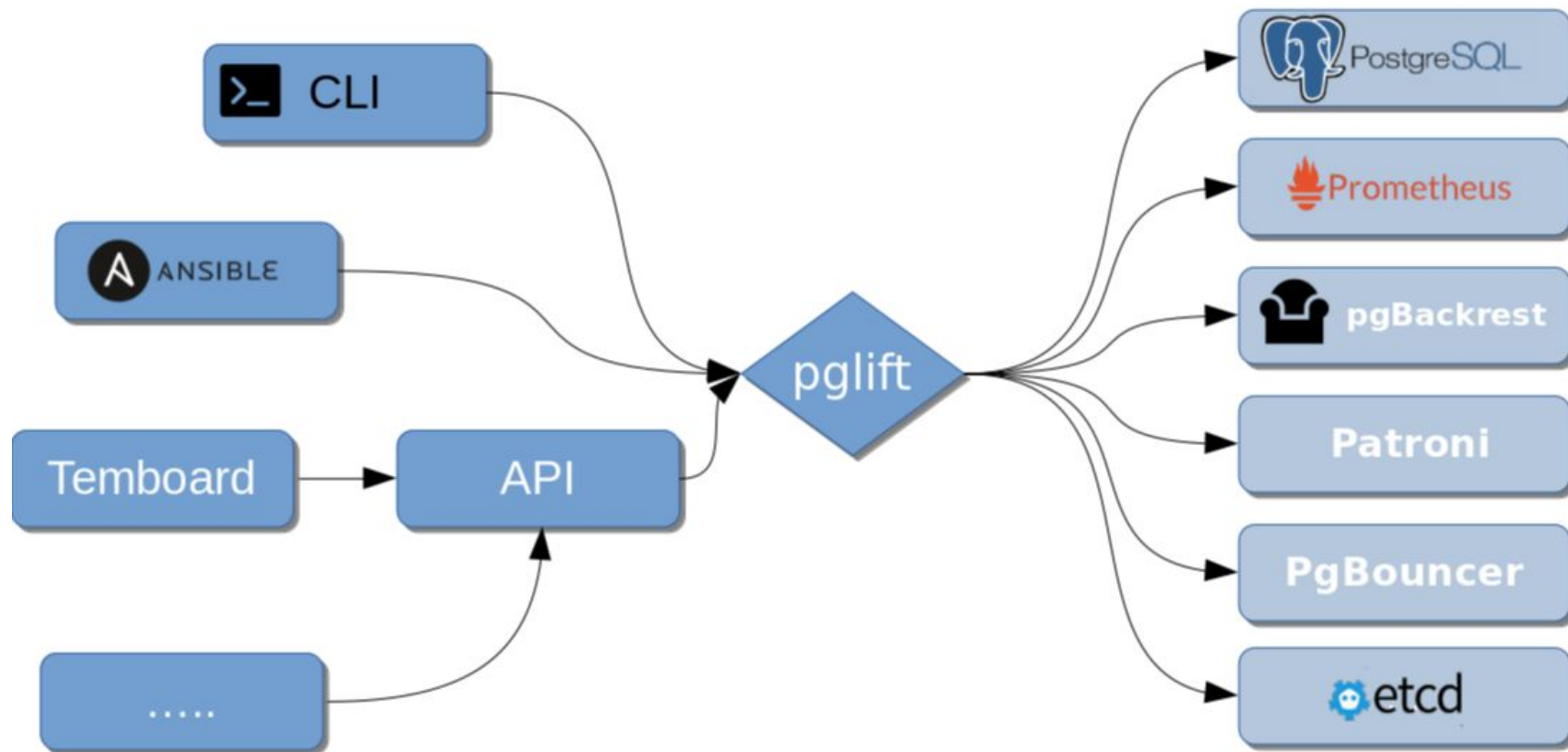
Cycle de Vie



Nouveautés du Socle 19

- Support de PostgreSQL 19
- Nouvelle Architecture des dépôts RPM
- 1 nouvelle API REST
- Focus Sécurité
- Chiffrement Transparent
- Opérateur K8S CNPG

Nouvelle API REST



Nouvelle API REST

DEMO

Nouvelle API REST

Nous sommes à votre écoute pour :

- Vous aider à interfacier vos outils à l'API
- Ajouter de nouveaux point d'entrée

Sécurité

Tous les services sont lancés en mode “user instance”

Les modules ansible fonctionnent sans privilèges root

Fonctionne parfaitement avec SELinux

Chiffrement TLS par certificats sur toute la stack

Interface avec un annuaire LDAP / AD

Sécurité

- Secured by design
- Open by default
- Réduire la surface d'attaque
- Séparation des Roles

Secured by design

Les vulnérabilités sont bien plus coûteuses et difficiles à corriger une fois un système déployé que si elles sont anticipées et éliminées dès les phases de conception du projet.

Pour PostgreSQL, cela implique de mobiliser des experts en amont et d'intégrer les mesures de sécurité dans les cadres de développement.

Par exemple, avec PostgreSQL Anonymizer, on peut rédiger la politique de masquage directement dans le modèle de données.

Open By Default

Consciente des limites de la « **sécurité par l'obscurité** » — c'est-à-dire faire reposer la sécurité d'un système sur le fait qu'un attaquant ne connaisse pas ses principes de fonctionnement — **l'ANSSI défend le principe d'« open-by-default »** (ouverture des codes sources et des données)

en veillant à ne pas exposer d'artefacts sensibles ou de configurations opérationnelles et en cohérence avec ses missions.

([Source](#))

Réduire la Surface d'Attaque

- Attack Surface Reduction (ASR)
- Limiter la circulation des données dans le SI
- Réduire le nombre de points d'entrée

Séparation des Rôles

Définir différents rôles avec **différentes politiques de masquages**.

Nommer les usages et les usagers

Notre Méthode

10 axes de sécurisation

- Une approche globale
- Valorisation les équipes techniques
- Une démarche itérative
- Un support pro-actif
- Outils 100% open-source

Axe 1 – Journalisation / Auditing

La journalisation est le socle de la preuve et de la détection.

Sans traces fiables, il est impossible de **prouver qu'une violation ne s'est pas produite**, ni de reconstituer une attaque après coup.

La complexité réside dans l'**équilibre entre exhaustivité** (tout tracer) **et performance** (ne pas saturer l'instance) ainsi que dans la durée de rétention légale des journaux.

⚠ NIS2 impose un contrôle d'accès strict avec journalisation horodatée des requêtes

Axe 2 – Isolation / SELinux

L'**isolation des processus et des environnements** est un rempart contre la propagation latérale: si un attaquant compromet un service, il ne doit pas pouvoir atteindre la base de données.

SELinux opère au niveau du noyau Linux avec des **politiques de contrôle d'accès obligatoire** (MAC), bien plus strictes que les permissions Unix classiques.

Les politiques SELinux sont notoirement **difficiles à écrire, maintenir et déboguer**. Surtout pour les instances PostgreSQL

⚠ NIS2 impose l'isolation système comme un prérequis de conformité

Axe 3 – Anonymisation

L'anonymisation est une obligation dès lors que des données personnelles circulent hors de l'environnement de production (recette, dev, analytics, IA).

Il est fondamentale d'avoir une **approche *privacy by design*** c'est à dire d'intégrer la politique de masquage dans le modèle de données.

⚠ Le RGPD demande une anonymisation robuste et documentée, pas symbolique.

Axe 4 – Gestion des Identités

La généralisation des “utilisateurs nommés” est une étape fondamentale et complémentaire de la journalisation.

Mais comment synchroniser l’annuaire interne (LDAP/AD) avec les instances PostgreSQL ?

⚠ NIS2 demande clairement une gestion stricte des droits et des contrôles d’accès.

Axe 5 – Chiffrement du Transit

Toute donnée circulant en clair sur le réseau — même interne — est interceptable.

TLS / SSL est la réponse standard mais **son intégration** avec la stack PostgreSQL **n'est pas triviale**.

⚠ DORA et PCIDSS ont des exigences claires en terme de chiffrement des communications.

Axe 6 – Chiffrement au Repos

Le chiffrement au repos (“Data At Rest”) est un mécanisme classique de protection des données.

Cela protège notamment **contre le vols des supports de stockage**.

Le Full Disk Encryption (FDE) est l’approche la plus courante et pleine compatible avec Postgres.

⚠ DORA et PCI-DSS établissent le chiffrement au repose comme une nécessité.

Axe 7 – Chiffrement à l'usage

Le chiffrement au repos n'est efficace que lorsque la base de données est éteinte, **ce qui n'arrive jamais**.

Le **chiffrement de la mémoire Postgres** ("shared_buffers") est l'approche **la plus sûre**.

⚠ Le chiffrement "data-in-use" n'est pas encore imposé par les différentes réglementations mais c'est une recommandation de plus en plus fréquente.

Axe 8 – Chiffrement applicatif

Le chiffrement applicatif (end-to-end encryption) est une approche garantissant une **granularité fine et une protection transversale**.

PostgreSQL fournit une boîte à outil permettant d'implémenter des applications qui intègre le **chiffrement au plus près de la donnée**.

⚠ Le chiffrement applicatif est une mesure complémentaire à toutes les autres couches de chiffrement

Axe 9 – Diagnostic Sécurité

Un système ne peut être sécurisé que si l'on sait dans quel état il se trouve.

Il existe des dizaines de **points de contrôle à vérifier**.

La complexité réside dans l'**automatisation des process de contrôles** et des **actions de remédiation**.

⚠ Le référentiel CIS Benchmark s'est imposé comme le standard du marché.

Axe 10 – Chaîne d'approvisionnement

La compromission de la chaîne logicielle est l'un des **vecteurs d'attaque à la plus forte croissance** (SolarWinds, XZ, etc...)

Pour un distributeur de PostgreSQL, cela implique :

- vérification de l'intégrité des paquets (signatures, checksums),
- traçabilité des composants (SBOM),
- garantie que les binaires livrés n'ont pas été altérés entre leur compilation et leur déploiement.

⚠ NIS2 impose un contrôle strict des prestataires et des "fournisseurs" de code.

Synthèse

#	Axe	Solution	Développé par
1	Journalisation	pgAudit	PGDG
2	Isolation / SELinux	pglift	DALIBO
3	Anonymization	PostgreSQL Anonymizer	DALIBO
4	Identités	ldap2pg	DALIBO
5	Chiffrement en Transit	TLS / SSL	PGDG
6	Chiffrement au Repos	FDE	-
7	Chiffrement à l'Usage	The Cryptic Elephant	DALIBO
8	Chiffrement Applicatif	pgcrypto	PGDG
9	Diagnostic	pgdsat	Hexacluster
10	Chaine d'Approvisionnement	delivery.dalibo.com	DALIBO

Aller Plus Loin avec DALIBO

- Workshop
- Diagnostic
- Demo TDE

Fins de comptabilité

Arrêt du support de PostgreSQL 14

Arrêt du support de Debian 22.04

Arrêt du support de Red Hat 8

Mais....

Possibilité de rester sur le Socle Postgres 18

Pistes de travail pour le socle 20

Intégration temboard - pglift

Database Branching

SBOM / inventaires des dépendances

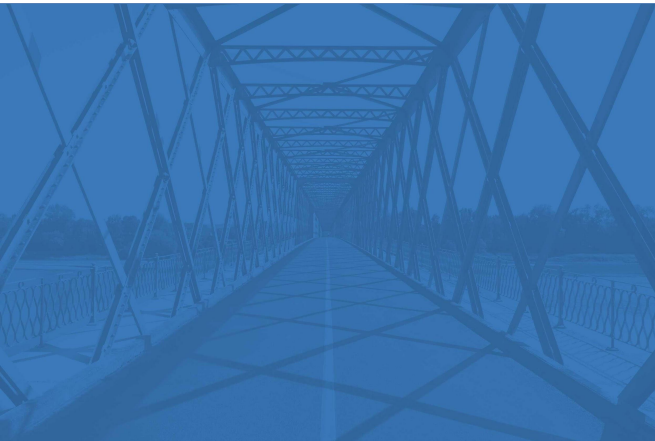
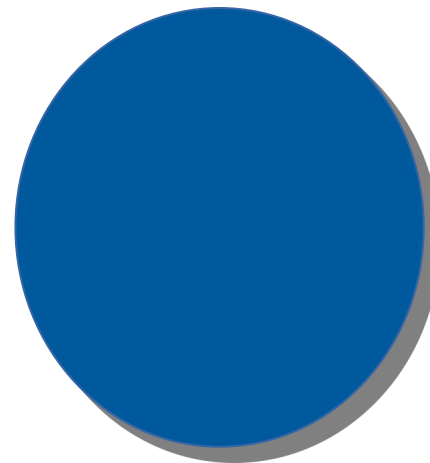
Sonde sql_exporter

pgcopydb

Damien Clochard

damien.clochard@dalibo.com

Product Owner Socle / 06 74 15 56 78



Notre prochain contact ?