

Un éléphant pour 50 000 étudiants dans une université

Nicolas Berne

Université Lumière Lyon 2



Qui suis-je ?

- Couteau-suisse : chef de projet, développeur, enseignant vacataire
- PostgreSQL ou une suite de rencontres (depuis mes études...)

Plan

- Pourquoi PostgreSQL dans la gestion des identités ?
- Comment l'utilisons-nous ?
- Retour d'expérience après quelques années

Gestion des identités

- Lier des données administratives et fonctionnelles avec des données techniques
- Identifier
- Authentifier
- Autoriser

Problématiques

- Environ 50 000 comptes utilisateurs actifs (étudiants, personnels)
- Synchronisation d'informations sur 3 annuaires (LDAP, AD) avec mot de passe identique
- Améliorer la complétude des annuaires (notamment SupAnn)
- Contraintes :
 - Création et suppression de 15 000 comptes étudiants / an
 - Jusqu'à 2 000 créations / jour
 - Pas de stockage du mot de passe
 - Financières
 - Exploitation minimale au quotidien
- Interruption de service la plus réduite possible

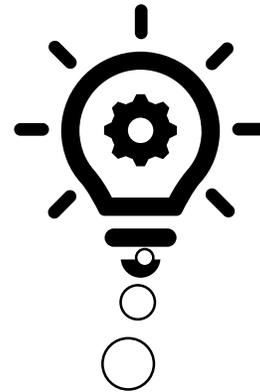
Historique

- Réflexion d'un passage d'alimentation d'**1** annuaire à partir d'une base Oracle/Java vers un autre système pour gérer **3** annuaires
- Travail collaboratif et réflexion avec notre ancienne DBA
 - ➔ Aller vers une autre DB que l'ancien système (exploitation au quotidien, coût licence)
 - ➔ Aller vers un langage de développement plus léger
 - ➔ Fiabilité et robustesse

PostgreSQL

Maturité de la solution

PG 7.X comme étudiant



Acquérir des compétences

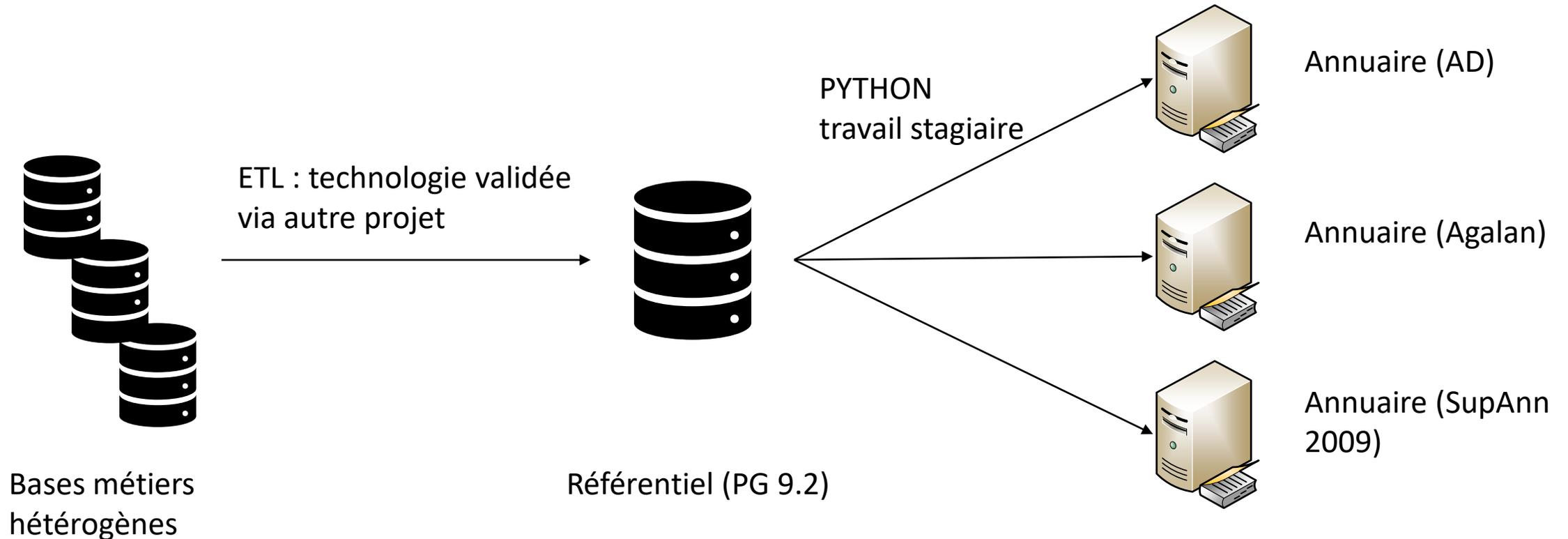
Support PL/Python

Essayer autre chose

Principes

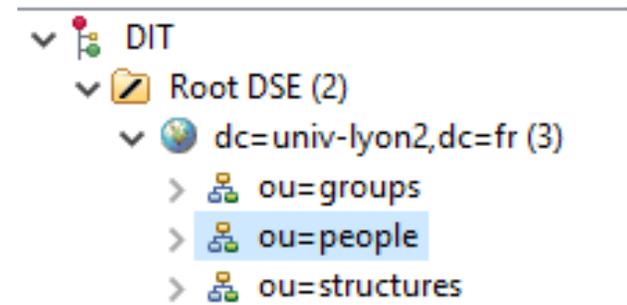
- Permettre une interopérabilité fine avec les annuaires LDAP
- Traiter au maximum les données côté DB pour éviter les A/R
- Processus de création de compte sensible
- Capacité à adopter une démarche « à la carte »

Architecture mise en place (Hiver 2017)



LDAP

- LDAP est un protocole
- Organisation hiérarchique des données sous-jacentes (structure arborescente)
- Feuilles peuvent se trouver à différents niveaux (Distinguished Name)
- Feuille contient des données à plat (attributs)
- Attributs peuvent être multivalués ou monovalués (un login unique, plusieurs adresses de messagerie)
- Gestion des groupes utilisateurs



DN: uid=tttest,ou=people,dc=univ-lyon2,dc=fr

Description d'attribut	Valeur
<i>objectClass</i>	<i>eduPerson (auxiliary)</i>
<i>objectClass</i>	<i>inetOrgPerson (structural)</i>
<i>objectClass</i>	<i>organizationalPerson (structural)</i>
<i>objectClass</i>	<i>person (structural)</i>
<i>objectClass</i>	<i>posixAccount (auxiliary)</i>
<i>objectClass</i>	<i>supannPerson (auxiliary)</i>
<i>objectClass</i>	<i>top (abstract)</i>
cn	Test Test
gidNumber	666545
homeDirectory	/user/tttest
sn	Test
uid	tttest
uidNumber	666545
displayName	Test Test
eduPersonAffiliation	affiliate
eduPersonPrimaryAffiliation	affiliate

Quelques particularités intéressantes

Expression régulière : qualité des données en amont

```
CHAINE :=  
REGEXP_REPLACE (CHAINE, '^ [^ [:lower:]]+ | [^ [:lower:  
]]+$', ' ', 'g');
```

Quelques particularités intéressantes (PL/Python)

```
CREATE OR REPLACE FUNCTION
profil_creation_compte_etape_init_login(i_id_compte
integer, i_login_character_varying)
  RETURNS integer
  LANGUAGE plpython3u
AS $function$
    import profil_compte
    return
profil_compte.creation_compte_etape_init_login(i_id_
compte, i_login)
$function$
;
```

Quelques particularités intéressantes (JSON)

- Gestion du format JSON très limitée en 9.2
- Migration en 12
- ➔ Nécessaire pour résolution problèmes techniques d'interrogation

```
SELECT data->'donneesPersonnelles'->>'prenom', data->
>'donneesPersonnelles'->>'nomUsuel'
from siham_individus
where data->'donneesPersonnelles'->>'matricule' =
'UN_MATRICULE'
```

- ➔ Nécessaire pour mieux « profiler » les individus (arborescence 117 profils fournie par Renater avec SupAnn 2021)

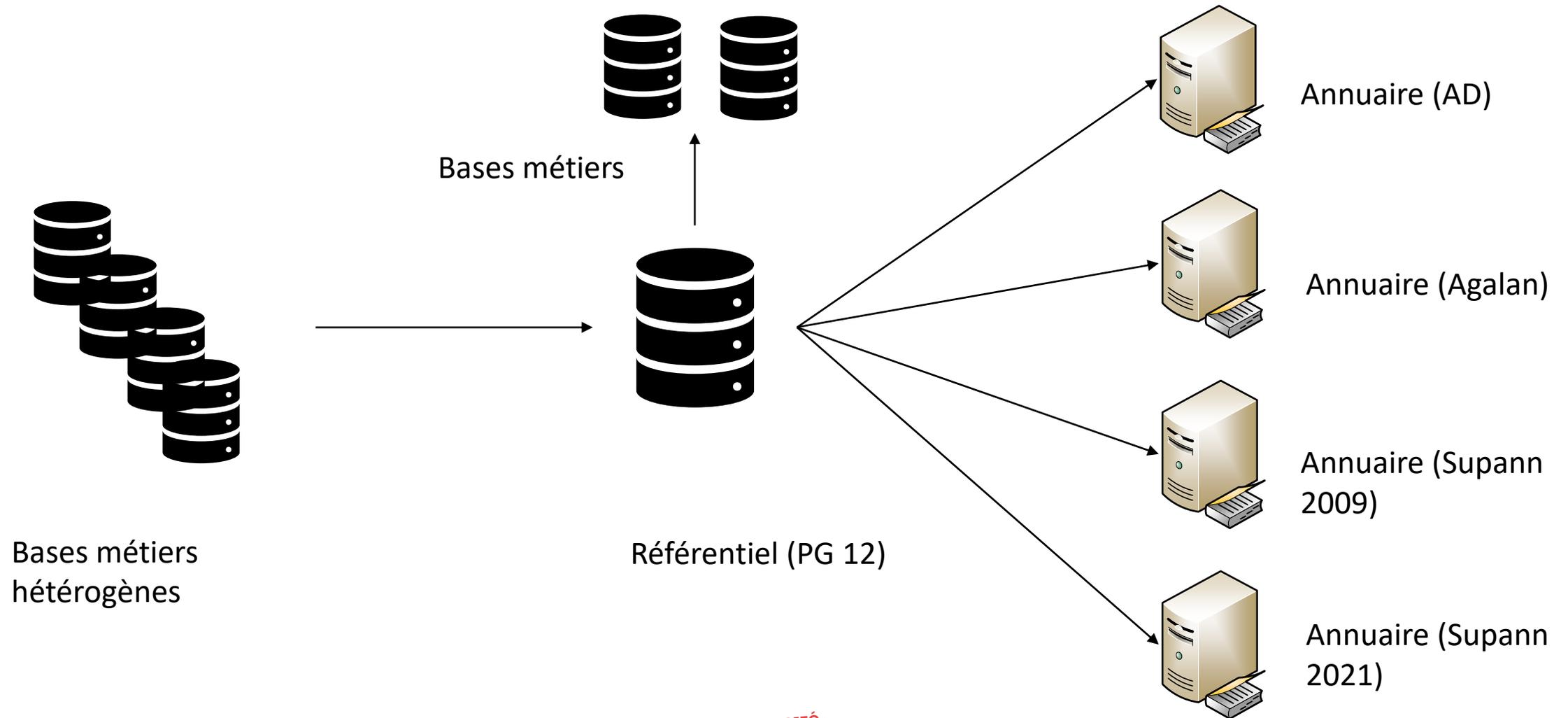
Retour d'expérience

Très positif car très fiable.

Un seul gros problème rencontré en 5 ans à cause d'une erreur humaine.

Réussite → Utilisation PostgreSQL / Python pour développement WEB (Django : ORM 😊) notamment gestion des élections universitaires.

Architecture en place (Hiver 2022)



Et si c'était à refaire ?

- 9.2 en 2017 proche de la fin sauf package propre à REDHAT → Mise à jour plus rapide (v12 et support JSON)
- KISS (éviter la suroptimisation inutile)
- Architecture robuste et PostgreSQL fait moins peur aujourd'hui

« La perfection est atteinte, non pas lorsqu'il n'y a plus rien à ajouter, mais lorsqu'il n'y a plus rien à retirer. » Antoine de Saint-Exupéry.

Remerciements

- À toutes celles et ceux qui ont permis de mettre en place cette solution fiable avec par ordre d'apparition :
 - Enseignante de maîtrise ;
 - DBA PostgreSQL (Clémentine) ;
 - Stagiaire Python (Mathilde) ;
 - DSI (Fabrice).
- Collègues de la DSI pour les évolutions et l'exploitation notamment Aurélien et Jérôme
- Dalibo (Laura) pour l'invitation à soumettre
- Mon employeur



Questions / réponses

Un éléphant pour 50 000 étudiants dans une université



PGSession 15

nicolas.berne@univ-lyon2.fr