eranea

Migration transparente de bases de données vers PostgreSQL

Savoir-faire d'Eranea

- Nous ciblons l'évolution technologique d'applications cœur de métier, dans le cadre de la transformation digitale des entreprises :
  - automatisation, iso-fonctionnalité et incrémentalité sont les points clés pour une migration transparente sans risque,
  - protection des investissements et de la propriété intellectuelle des clients.

- Nous sommes aussi actifs dans le domaine des migrations de grands systèmes vers Linux et avons développé un jeu complet d'outils et de technologies autour de :
  - analyse de code (Cobol, Java, SQL, ...),
  - transformation de code,
  - analyse de données et outils de tests,
  - chaînes DevOps pour l'automatisation.

- Ces outils / technologies de migration permettent de réaliser la migration transparente pour les applications de base de données X vers PostgreSQL.

- Nous possédons 100% du code source de ces outils et pouvons donc les adapter ou les étendre.

Migration de bases de données

Pourquoi migrer ?

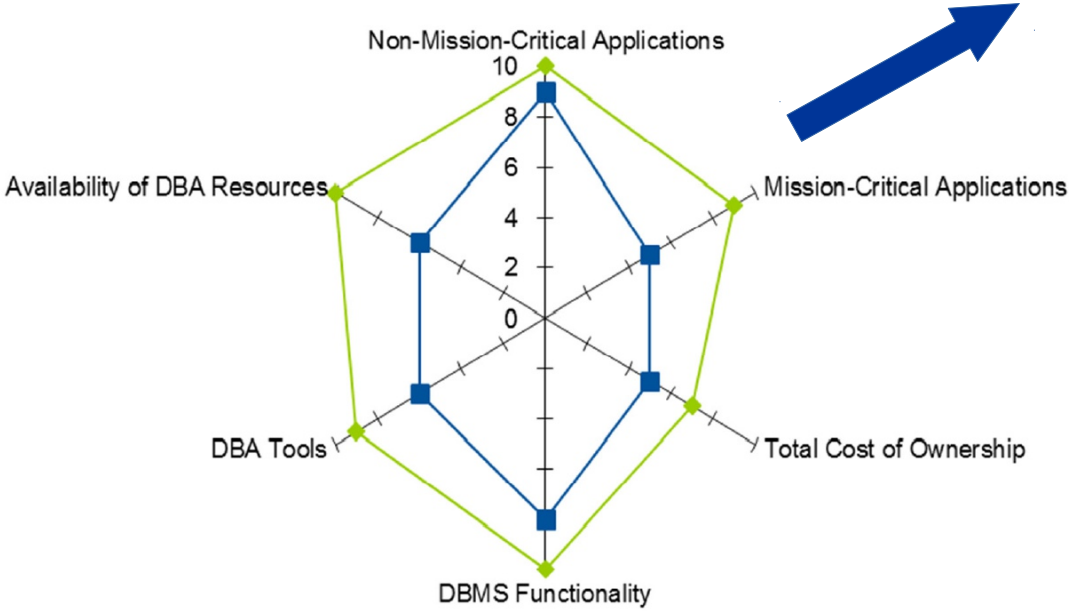
- Les grands comptes utilisent différentes bases de données, mais ils souhaitent :
  - simplifier leur architecture et leurs opérations,
  - harmoniser leurs systèmes actuels,
  - réduire leur coûts IT.

- Ils veulent réduire les coûts de licence :
  - en obtenant de meilleures rabais grâce à de plus gros volumes,
  - en utilisant le levier Open Source. Pour Gartner (2015), les DBs OSS sont matures pour les applications critiques,
  - le modèle de licence des éditeurs traditionnels est trop strict/coûteux dans un contexte virtualisé (cloud, ...).

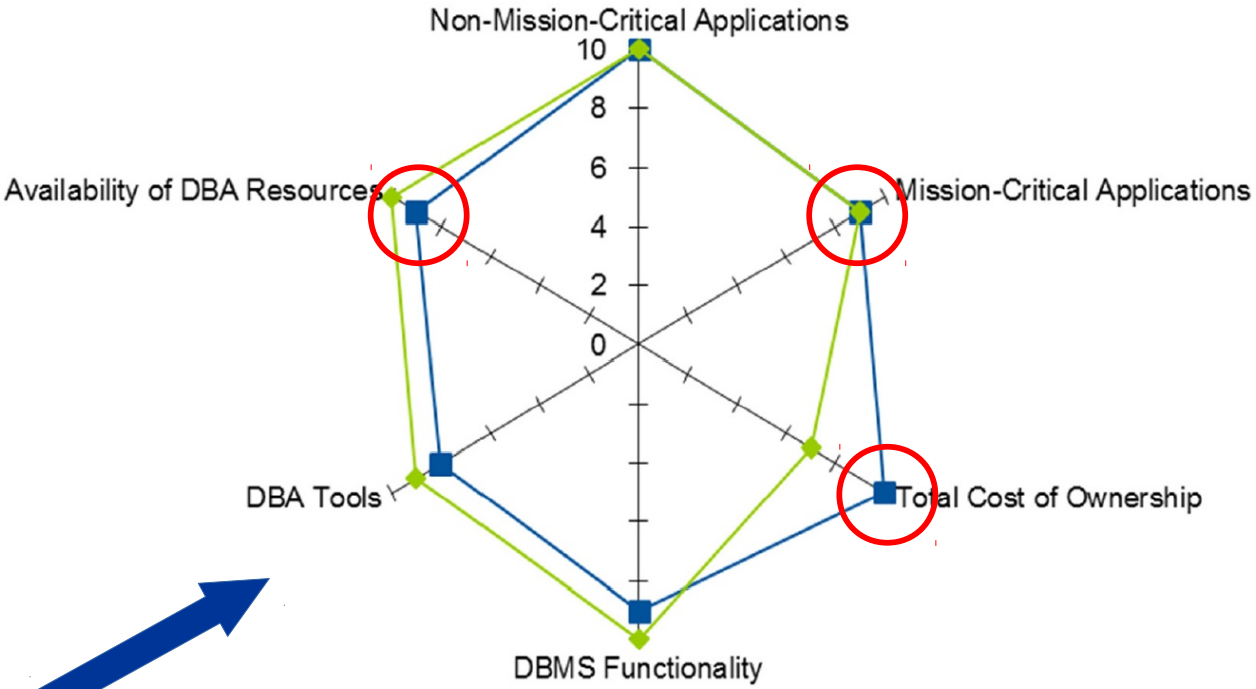
eranea

Maturité des DBs OSS

DBMS Maturity Evaluation
Source: Gartner 2015



2009



2015

Points durs

- SQL est un standard, mais :
  - il n'est pas totalement spécifié : il reste de nombreuses possibilités d'interprétations,
  - des différences / incompatibilités existent : syntaxiques, sémantiques, représentation des données, format des réponses, ...

- Le code source des applications doit évoluer lors d'une migration afin de prendre en compte les changements :
  - beaucoup de travail coûteux sans évolution fonctionnelle
  - les besoins du business seront toujours prioritaires,
  - cela requiert de gros efforts de coordination pour implémenter et tester simultanément tous les changements et les évolutions applicatives,
  - certaines différences sont complexes à pallier (arrondis, formats, ...).

- Les modifications distribuées dans le code sont la première barrière (et la plus haute) au changement de base de données :
  - beaucoup trop d'efforts même pour un PoC !
  - les clients restent où ils sont avec ce qu'ils ont ...

eranea

Retour d'expérience

Points importants :

- Justesse et efficacité de la conversion des ordres SQL.
- Obtenir la majorité des requêtes SQL.
- Avoir la couverture de tests la plus large possible.
- Pouvoir rejouer les tests sur la DB cible, sans dépendre des applications.
- Cas potentiellement complexes de certains ordres JDBC très souples, par exemple PreparedStatement.setObject.
- Gestion des calculs / arrondis.
- Collating sequences.

eranea

Notre expérience

- Ne changez pas une seule ligne de code !
- Nous centralisons les adaptations en un seul point hors de l'application, via un "Proxy Driver", qui adapte les requêtes et réponses en temps réel.
- Nous adoptons une approche iso-fonctionnelle

  ⟶ cela rend les migrations de DBs transparentes pour les applications,
  ⟶ la migration est réussie lorsque les résultats avant et après sont identiques.
- Les changements seront reportés dans l'application après la migration.

eranea

Avantages

- **Pas d'impact business :** les développeurs restent concentrés sur la maintenance fonctionnelle, qui n'est pas bloquée pendant la migration.

- **Efficacité :** tous les changements sont faits à un seul endroit, avec conversion à la volée de toutes les requêtes SQL, y compris celles construites dynamiquement, ainsi que les formats de données, de résultats, ...

- **Incrémentalité :** Migration incrémentale par schémas indépendants possible (mode hybride), support XA nécessaire.

- **Mesurabilité, contrôle :** Le "Proxy Driver" peut être utilisé pour mesurer / comparer la qualité et les performances des transformations.

- **Décommissionnement :** Les changements seront reportés dans l'application après la migration lors des évolutions applicatives pour le business.

Composants de migration

- Convertisseur de DDL : Utilisé pour convertir automatiquement les DDL de la DB source vers la DB cible. Crée les objets dans la DB cible avec la syntaxe correcte (tables, vues, indexes, contraintes, clés, droits, ...)

- Service automatique et efficace de déchargement et rechargement / adaptations des données de la DB source dans la DB cible.

- Proxy Driver avec les fonctions suivantes :
  - driver JDBC XA complet,
  - transformation des requêtes SQL (SQL de la DB source vers SQL de la DB cible) et des réponses en temps réel,
  - support de XA pour les transactions distribuées et le mode hybride (accès sur DBs source et cible).

eranea

Composants pour les tests

- Proxy Driver avec fonctionnalités supplémentaires :
  - capture et archivage de l'intégralité de l'activité JDBC des applications sur une base de données (requêtes SQL, réponses, ...),
  - mode passif (= transparent) pour la capture sur la DB source,
  - mode actif avec transformation des requêtes SQL et des réponses en temps réel sur la DB cible,
  - mesure des performances pour comparaison des DBs source et cible.

- Système de ré-exécution / comparaison des captures JDBC de la DB source sur la DB cible.

- Service de capture centralisée des changements de données ("Change Data Capture" ou CDC) :
  - capture d'un ensemble de triggers ajoutés sur toutes les tables, utilisés pour détecter et persister tous les changements sur les données.
  - système de comparaison auto-adaptative des valeurs de triggers.

- Mesure objective de la qualité de la migration par analyse de la couverture de code.

eranea

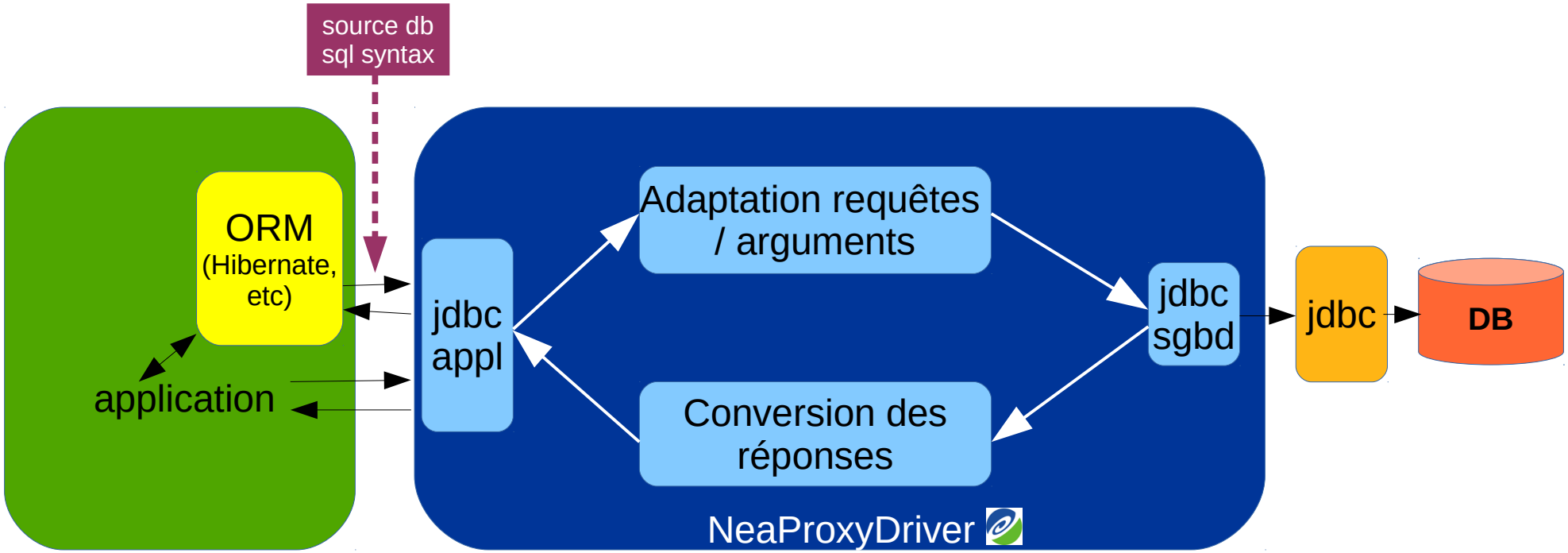
PostgreSQL

- Avantages :
  - proximité avec DB/2,
  - possibilité de création en C de types de colonnes étendus et de fonctions émulant le système source (echar, evarchar, ...),
  - souplesse des collating sequence,
  - richesse de la base de données.

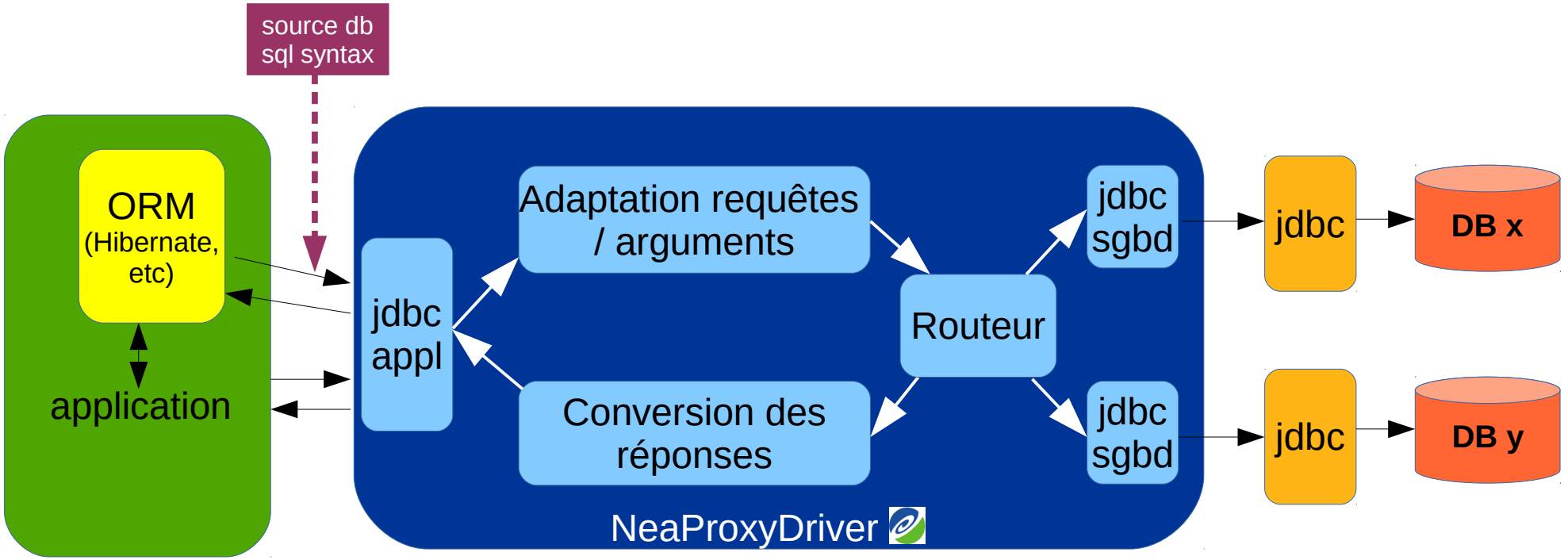
- Limitations actuelles :
  - les 0 binaires dans les colonnes textuelles ne sont pas supportés,
  - tailles des colonnes bytea et text limitées à 1Go

eranea

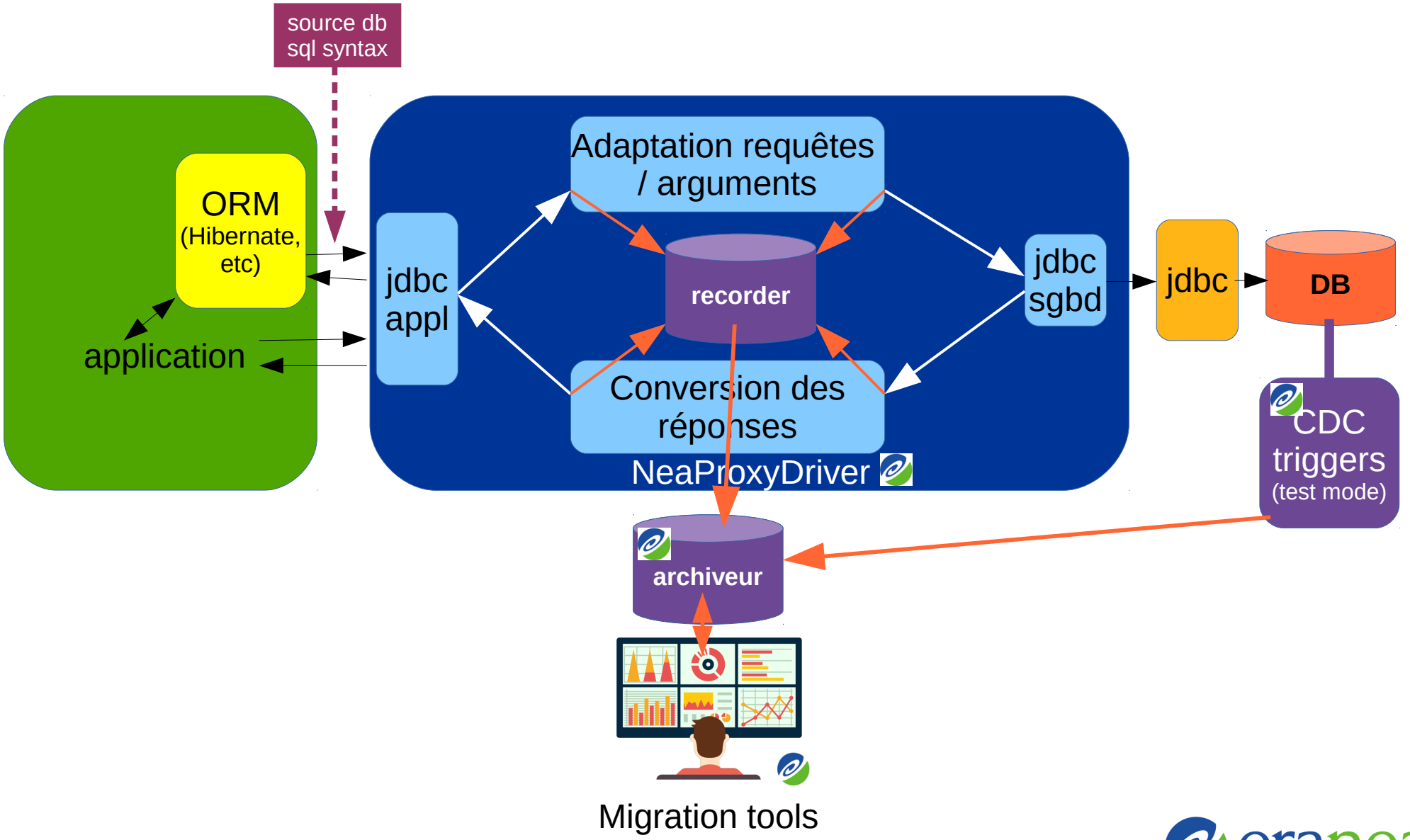
Proxy Driver



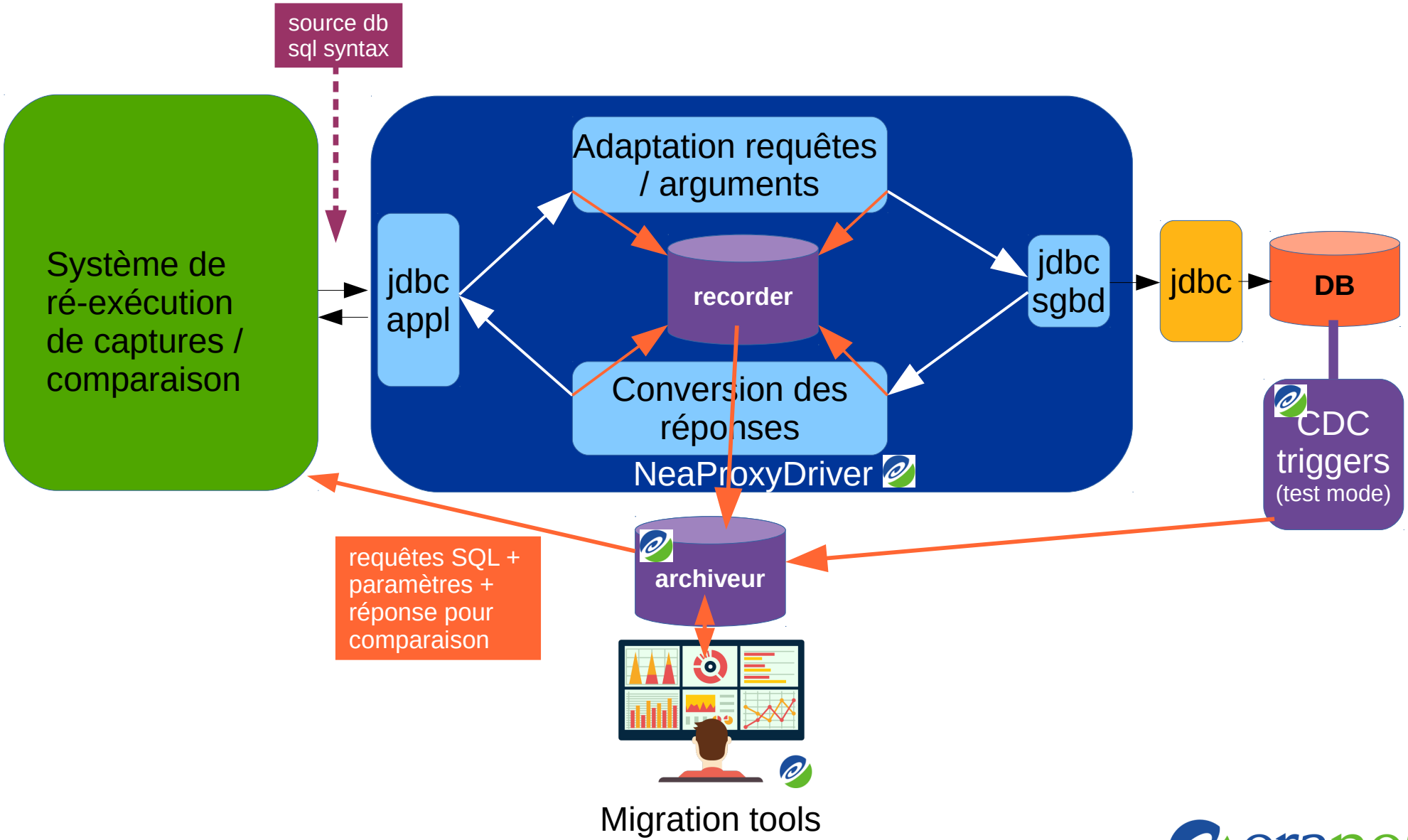
Proxy Driver / Mode hybride



Proxy Driver avec capture



Ré-exécution de captures



Migration tools

Réalisation de la migration

Etapes de la migration - préparation

- Analyse du contexte :
  - workshops / questionnaires,
  - définition des campagnes de tests.
- Obtention des DDL de la DB source :
  - conversion des DDL au format de la DB cible.
- Installation d'un environnement source de référence.
- Installation d'un environnement cible pour les tests.
- Installation du Proxy Driver en mode passif sur les différents environnements (dev, intégration, production) pour collecte des captures JDBC pendant X semaines, puis analyse pour déterminer le plan de migration de la production.

eranea

Etapes de la migration - tests

- Création de la structure de la DB cible avec les DDL modifiés.
- Déchargement de la DB source / rechargement des données dans la DB cible.
- Captures des tests applicatifs avec Proxy Driver en mode passif sur la DB source.
- Ré-exécution des captures sur la DB cible avec Proxy Driver en mode actif :
  - comparaison des résultats attendus, y compris des valeurs de triggers.
- Validation finale avec la DB cible :
  - exécution des scénarios de test avec le Proxy Driver en mode actif sur la DB cible,
  - comparaison des résultats attendus, y compris des valeurs de triggers.
  - prise en compte des aspects de performances.

eranea

Etapes de la migration - finalisation

- Mise en place des nouvelles infrastructures
  - hardware, logiciels, haute disponibilité, backup, ...
- Formation des DBAs à la DB cible
  - Mise en œuvre des nouvelles procédures d'administration : diff / full backup, log archiving, reorg, runstats, restore, etc.
- Définition du plan de mise en production :
  - migration incrémentale ou complète ?
  - Planification des différentes migrations, par environnement (dev, integration, prod)
- Planification du transfert des données dans la DB cible de production.
- Exécution des migrations, par environnement de criticité croissante.

eranea

Post migration

- Suppression du Proxy Driver :
  - modifier les applications avec changements de syntaxe, de paramètres et de réponses fournies par les logs du Proxy Driver,
  - utiliser le mode hybride pour mettre le Proxy Driver partiellement en mode passif (=transparent), pour les applications corrigées,
  - dé-commissionnement du Proxy Driver quand les applications ont été complètement corrigées

eranea

Merci !

questions ?

Contact:

Didier DURAND

didier.durand@eranea.com

+ 41 79 944 37 10