Microsoft®
tech·days
2012

palais des
congrès
Paris

7, 8 et 9
février 2012

# La Sécurité avec SQL Server

7 février 2012

Jean-Pierre Riehl
MVP SQL Server
AZEO
Jean-Pierre.Riehl@azeo.com

Pascale Doz
Consultante SQL Server
Indépendante
pascale@pascale-doz.com

# AZEO / Pascale DOZ

- AZEO

- Pascale Doz
  - Consultante SQL Server depuis la version 4.2
  - Architecture, Accompagnement, Conseil, Formation

# Agenda

- Back to Basics
- Contrôle d’accès
- Protection des données
- Surveillance
- SQL Server 2012

# Back to Basics

# Back to Basics

« La sécurité dépend du risque »

« La sécurité est alignée sur le maillon le plus faible »

« Principe du moindre-privilège »

# Back to Basics

## A retenir

Accès

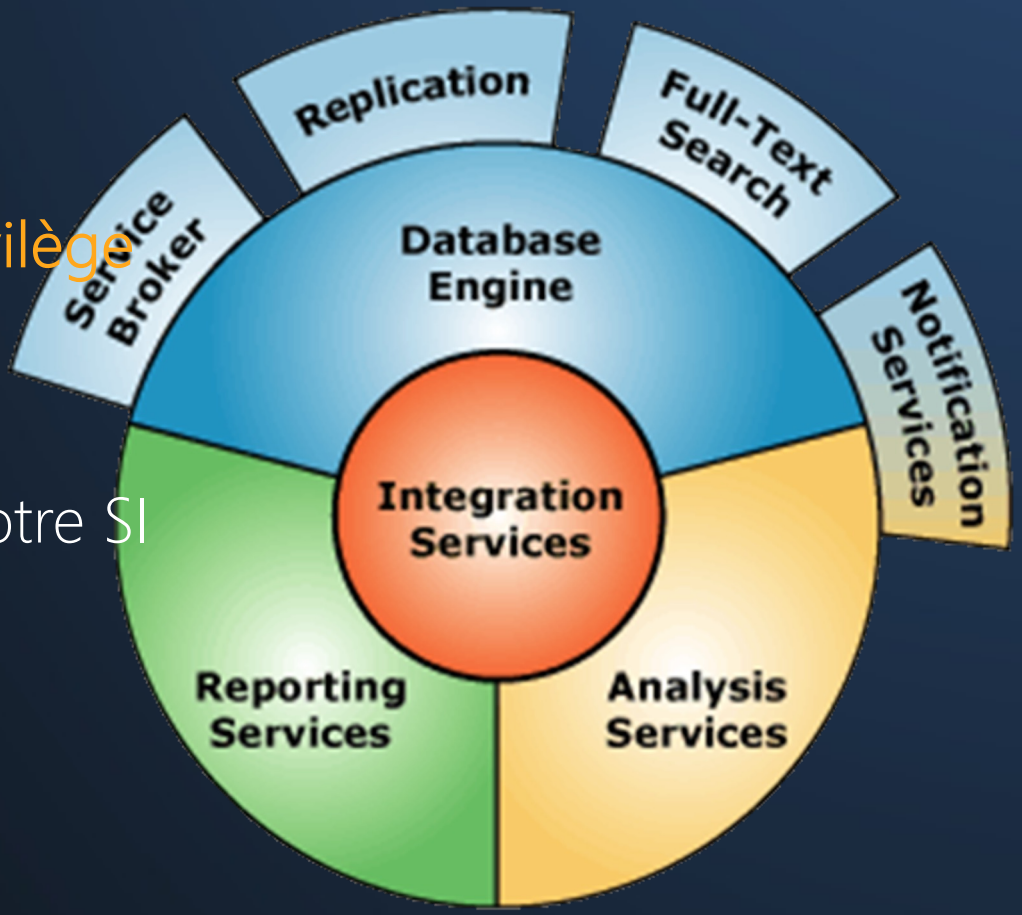
Protection

Surveillance

# Accéder

# Surface d'exposition

- Limiter les ouvertures
  - Composants
  - Endpoints
- Principe du moindre privilège
- Pensez à l'ensemble de votre SI



# Accéder à SQL Server

## Principals

- Login vs User

- A la base, il y a toujours un Login
  - Mode intégré ou mixte
  - Permet de se connecter à l'instance SQL

- Le User/Certificat permet d'utiliser la base de données
  - Donne un contexte au login
  - *Attention aux utilisateurs guest / dbo*

- Rôles (serveur ou base de données)

# Authentification

## La bonne pratique :

- Utilisation de la sécurité Windows
  - Gestion centralisée avec Active Directory
  - Respect des politiques d'entreprise
  - Pas de gestion de mot de passe

- Mais toujours un besoin de la sécurité SQL pour les « clients » non-windows

# Accéder à SQL Server

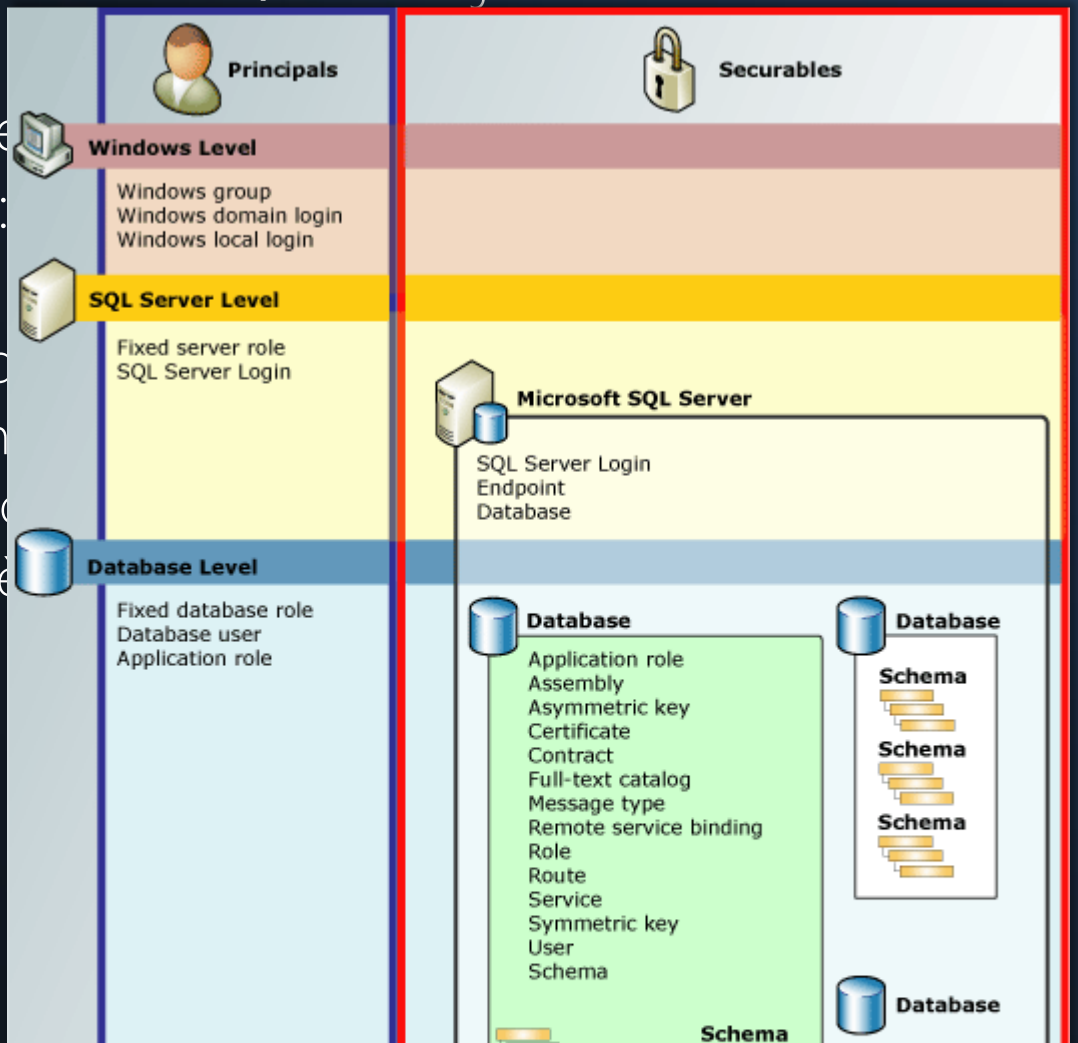
## Securables

- De l'instance SQL aux objets

- On donne
  - GDR :

- Principe c
  1. Défin
  2. Echec
  3. Succè

Attention au
Attention au



# Accéder à SQL Server

- Quelques bonnes pratiques :
  - Authentification intégrée, politique de mot de passe
  - Pas d'utilisation de dbo ou sysadmin
  - Utilisation des SCHEMAS
  - Cloisonnement
  - Principe du moindre privilège

# Protéger

# Une histoire de clés

**Service Master Key** : niveau instance

**Database Master Key** : niveau base de donn

Clés asymétriques

Certificats

Clés symétriques

**Importance des sauvegardes !**

# Une histoire de clés

**Extensible Key Management : EKM**

- Nouveauté de SQL Server 2008
- Exportation de la gestion des clés dans une appliance dédiée
  - Hardware Security Module (HSM)

**Ex : Arx PrivateServer, Thales nCipher, SafeNet Luna**

# Chiffrement des données

- Ne remplace pas la politique d'autorisations
- Protection « *In-depth* »

- Différentes granularités
  - **Serveur** : géré par Windows (EFS ou BitLocker)
  - **Base de données** : Transparent Data Encryption
  - **Données** : fonctions de chiffrement
- Différents algorithmes : *DES, TRIPLE_DES, AES (128, 192, 256)*

- Attention au canal de communication

# Chiffrement des données

## Transparent Data Encryption

- Pas d'action de l'utilisateur : transparent
- Chiffrement des fichiers MDF et LDF
- Chiffrement dans TempDB
- Chiffrement des backups
- Les buffers en mémoire ne sont pas chiffrés

- Impact en performance : 3 à 5% (sur un TPC-C) mais 28% de CPU en plus

# Chiffrement des données

## Transparent Data Encryption

```
--création de la DEK
CREATE DATABASE ENCRYPTION KEY
 WITH ALGORITHM = AES_128
 ENCRYPTION BY SERVER CERTIFICATE aCert;

--activation de l'option
ALTER DATABASE maBase
 SET ENCRYPTION ON;
```

# Chiffrement des données

## Cell-Level Encryption

- Chiffrement explicite de l'utilisateur
  - Impact sur les applications
- Granularité fine
- Attention aux « *dictionnary-attack* »

- Utilisation des fonctions : EncryptBy, DecryptBy, SignBy, VerifySignedBy, etc.

# SQL Injection

- C'est l'attaque la plus fréquente
- Pour s'en protéger :
  - Politique d'autorisation
    - Principe du moindre privilège
  - Proscrire le SQL Dynamique
  - Utiliser des requêtes paramétrées
    - Utilisation de SqlParameter
  - Contrôler les saisies
    - Utilisation des regexp
    - Utilisation de QUOTENAME(), REPLACE()

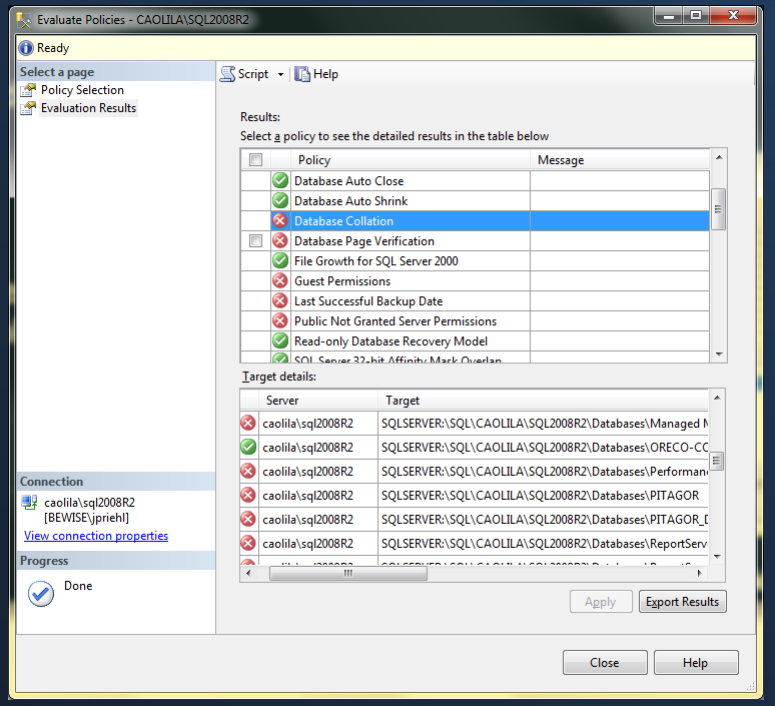
# Surveiller

# Policy-based Management

Policy :

→ Evaluer des Conditions sur les Facets du serveur

- Possibilité de les évaluer
  - À la demande
  - Planifiée
  - On Change



# DDL Triggers

- Triggers sur les instructions DDL
  - CREATE, ALTER, DROP, etc.
- Scope Base de données ou Serveur
- Transactionnel
- Permet de rajouter du code :
  - Annulation de la transaction
  - Application de règles métier
- Event Notification (basé sur service Broker / asynchrone)

# AUDIT

- Permet de tracer ce qu'il se passe sur un serveur
- Distinction entre le lieu ou on trace et ce qu'on trace
  - CREATE SERVER AUDIT
  - CREATE  XXX  AUDIT SPECIFICATION
- Granularité fine
  - Limitation à des utilisateurs particuliers
  - Limitation à un scope particulier (objets, actions)
- Basé sur le moteur d'événements
- Plus performant que la TRACE

# Events et TRACE

- Le système d'événements est à la base des fonctionnalités de monitoring

- On peut utiliser les événements pour réagir dans « l'application »
  - Traitement des événements via le Service Broker

```
CREATE EVENT NOTIFICATION ev ON SERVER
FOR CREATE_DATABASE
TO SERVICE 'serv
```

- La TRACE est optimisée pour ne pas perturber l'activité
  - Pas de packaging XML
- La TRACE est exploitée en dehors de SQL Server

# Et 2012 ?

# Nouveauté !

## Contained Databases

- Séparation du lien avec l'instance
- Isolation de la base
  - Plus besoin de LOGIN pour les utilisateurs

- Utilisation
  - Pratique pour la mise en place du mirroring (AlwaysOn)
  - Pratique pour les éditeurs

# Conclusion

# Conclusion

- Principe du moindre privilège
- Audit régulier (temps réel ☺)
- Pensez la sécurité au niveau du SI

# Mais encore…

Analysis Services

Data Quality Services

Power Pivot

Integration Services

Service Broker

Réplication

Log Shipping

Reporting Services

AlwaysOn

Mirroring

Master Data Services