

Synchroniser ses données

C'est plus pas facile que c'est compliqué !



Merci à nos sponsors...



PYRAMID
ANALYTICS



Microsoft



Cellenza



Maltem
Consulting Group



effektor
SELF SERVICE DATA WAREHOUSE & BI



SUPINFO
International University



SQLSaturday Paris 2015



Merci à nos volontaires...



SQLSaturday Paris 2015

GUSS



PASS **SQL**
saturday



Arian Papillon



a.papillon@datafly.fr

<http://www.datafly.fr>

<http://blog.datafly.pro>

<http://mssql.fr>



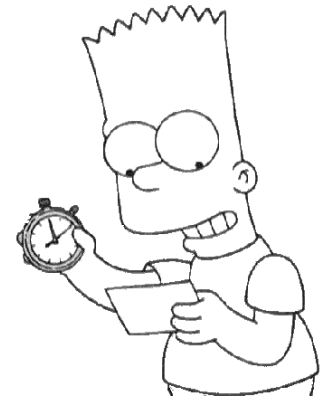
mssql.fr

L'actualité technique MS SQL Server en France (et ailleurs)



Synchroniser ses données

- Synchronisons nos montres : nous avons une heure pour parcourir un sujet qui mérite au moins la journée...
 - Pourquoi et comment
 - Ce qu'il y a sous le capot
 - Change Tracking
 - Change Data Capture
 - Service Broker
 - .. Et la réplication ?



Pourquoi faire

- Avoir des données identiques dans plusieurs tables/bases/instances/serveurs
 - Datawarehouses et bases de reporting
 - Bases de sites web
 - Applications multiples
 - Données distribuées
 - ...
- Objectif : synchronisation incrémentielle
 - Répercuter seulement ce qui a changé



Quelles sont nos contraintes et besoins

- Consommation des réplicas :
 - Base de secours seulement pour HA/DR ?
 - Lecture seulement ?
 - Mise à jour possible (des mêmes lignes) et gestion des conflits ? (synchronisation bi-directionnelle)
- Hétérogénéité
 - Structures différentes ?
 - Réplicas non SQL Server ?
- Fréquence de synchronisation
 - Périodique ou au fil de l'eau?
- Quelle édition de SQL Server ?



Comment conserver ses données synchronisées ?

- Après la première synchronisation :

Identifier les modifications

Se servir des dates de mise à jour

Historiser dans une table

Capturer les transactions dans le journal

Transmettre un message au destinataire



Appliquer les modifications

Avec TSQL (insert, update, delete)

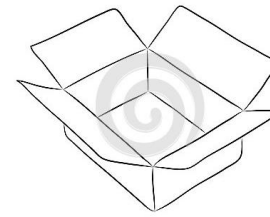
Avec SSIS

Avec une tâche de réplication (TSQL aussi)

Traiter un message reçu



Ce qu'il y a dans la boîte



- Change Tracking : suivi des modifications
- Change Data Capture : capture de données modifiées
- Service Broker : envoi de messages entre services
- Réplication : boîte à outils avec plusieurs méthodes de synchronisation (*mais ce n'est pas pour aujourd'hui, faute de temps*)

- SSIS : l'ETL pour le transport

- + Log Shipping, Database Mirroring, AlwaysOn : pour la haute disponibilité et au delà...





Suivi intégré au moteur : finis les triggers de journalisation

Change Tracking



Change Tracking (CT) – suivi des modifications

- Permet aux applications d'identifier quelles lignes d'une table ont été mises à jour (par insert, update ou delete) depuis la dernière synchronisation.
 - Ne conserve pas les différentes versions d'une même ligne
 - Tracke la PRIMARY KEY, la table surveillée doit en avoir une
 - Suivi SYNCHRONE : l'alimentation de la table de tracking fait partie du plan d'exécution
- Scénario : synchronisation périodique, vers SQL Server ou non
- Editions : disponible dès l'édition standard
- Sélection et transport des données avec TSQL ou SSIS



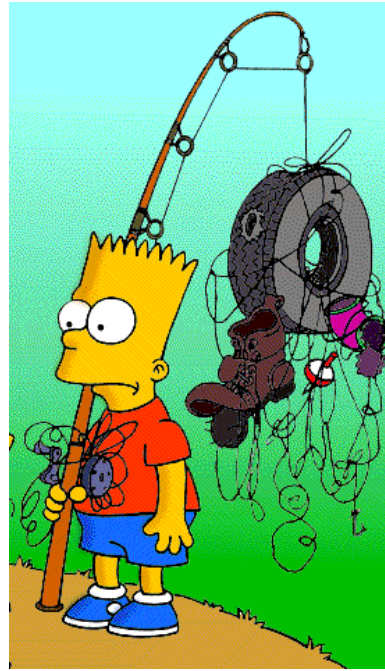
Demo



CT : Mise en œuvre

- Activer le CT sur la base
 - `ALTER DATABASE MyDB SET CHANGE_TRACKING= ON (CHANGE_RETENTION = 2 DAYS, AUTO_CLEANUP = ON)`
- Activer le CT sur chaque table concernée
 - `ALTER TABLE MyTable ENABLE CHANGE_TRACKING WITH (TRACK_COLUMNS_UPDATED = ON)`
- Conserver le numéro de version actuelle
 - `SELECT @s = CHANGE_TRACKING_CURRENT_VERSION()`
- Extraire les lignes modifiées depuis ce moment
 - `SELECT * FROM CHANGETABLE(CHANGES MyTable, @s) CT`
 - Tester les colonnes modifiées avec `CHANGE_TRACKING_IS_COLUMN_IN_MASK()`





Capturer dans le journal de transactions : comme la réplication transactionnelle

Change Data Capture



Change Data Capture (CDC) : capture des données modifiées

- Capture toutes les activités de mise à jour d'une table (insert, update, delete) et fournit l'historique détaillé des changements aux applications.
 - Conserve les différentes versions
 - Suivi ASYNCHRONE par un job de « log reader » exécuté par l'Agent SQL
- Scénario : synchronisation périodique, vers SQL Server ou non
- Editions : disponible en édition Entreprise (et développeur ou évaluation)
- Sélection et transport des données par TSQL ou SSIS : composants SSIS spécifiques



Demo

- TSQL



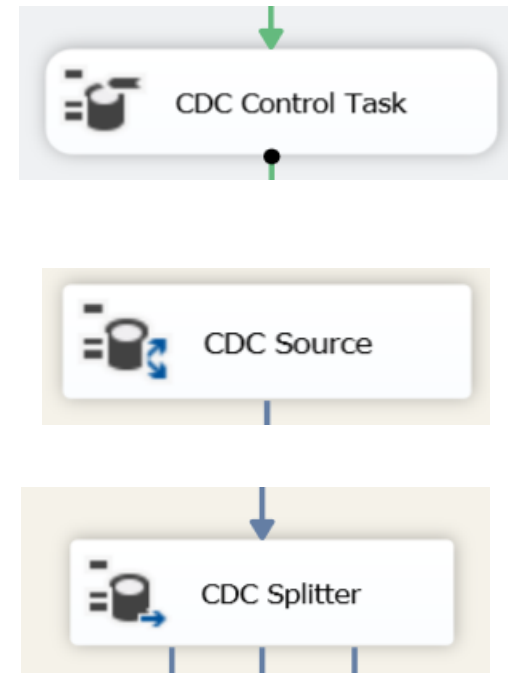
CDC : mise en oeuvre

- Activation du CDC :
 - Dans la base : `EXEC sys.sp_cdc_enable_db`
 - Sur chaque table : `EXEC sys.sp_cdc_enable_table`
`@source_schema = N'dbo', @source_name = N'MyTable',`
`@role_name = NULL, ...`
- Localiser les lsn de début et de fin
 - `sys.fn_cdc_get_min_lsn('CDCInstance')`
 - `sys.fn_cdc_get_max_lsn()`
 - `sys.fn_cdc_map_time_to_lsn('largest less than or equal',@datevar)`
- Extraire les modifications
 - `cdc.fn_cdc_get_all_changes_CDCInstance(@from_lsn, @to_lsn, '...')`
 - `cdc.fn_cdc_get_net_changes_CDCInstance(@from_lsn, @to_lsn, '...')`



CDC : composants SSIS

- **CDC Control Task** : gestion de l'état de synchronisation et des plages de LSN
 - ILSTART, ILEND, ILUPDATE
 - TFSTART, TFEND, TFREDO
 - ERROR
- **CDC Source** : extrait les données du CDC
- **CDC Splitter** : sépare les données modifiées en 3 flux : INSERT, DELETE et UPDATE



Demo

- Avec SSIS





Communiquer par messages entre bases de données

Service Broker

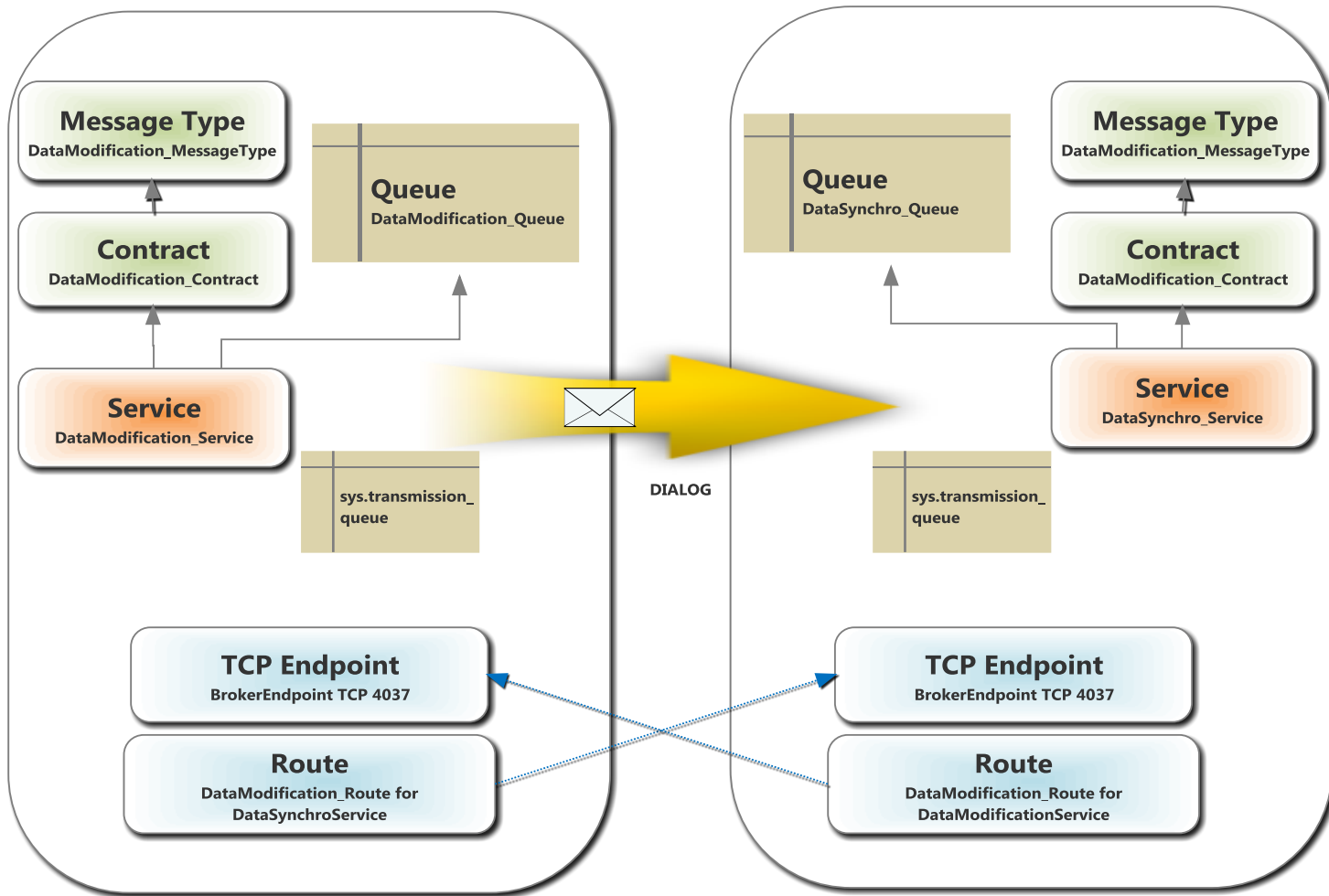


Service Broker

- Middleware orienté message (MOM) : outil de messagerie de base de données permettant de gérer des flux de données de manière asynchrone entre bases ou serveurs SQL Server.
 - Déjà utilisé en interne par Database Mail, Query Notification et les notifications d'évènements
 - Utilisable pour toutes sortes de scénarios de données distribuées
- Scénario : synchronisation de bases de données SQL Server différentes, via des messages transmis entre bases ou instances
- Editions : disponible dès l'édition standard
- Mise en œuvre avec TSQL



Fonctionnement de Service Broker



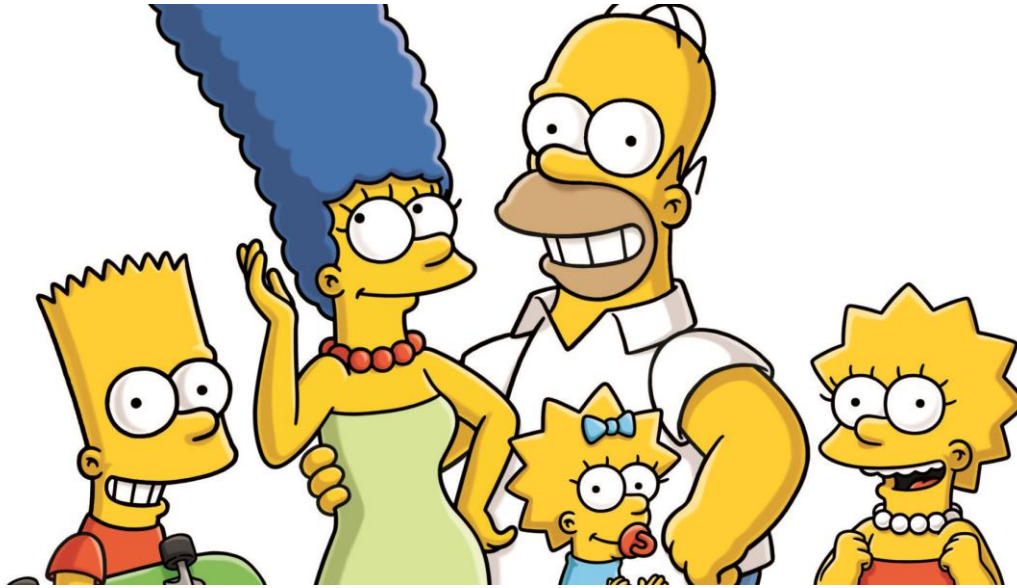
Demo



Mise en oeuvre

- Activer : ALTER DATABASE SET ENABLE_BROKER
- Créer les objets : message types, contracts, queues, services, endpoints, routes
- Initialiser la conversation et envoyer :
 - BEGIN DIALOG CONVERSATION FROM SERVICE TO 'SERVICE'...
 - SEND ON CONVERSATION
- Recevoir : RECEIVE
- Terminer la conversation : END CONVERSATION





Plusieurs méthodes pour synchroniser ses données

Et la réplication ?



La réplication

- Un ensemble de technologies pour copier, distribuer, synchroniser des données (articles) entre bases
 - Basées sur un modèle à 3 rôles : éditeur / distributeur / abonné
 - Utilise des agents de réplication (exécutables lancés par l'agent SQL Server)
 - Configurable par assistant (ou en TSQL)
- Editions : disponible en édition Standard (sauf la réplication peer to peer)
- Scénarios : synchronisation périodique ou au fil de l'eau (transactionnelle et fusion)
 - vers SQL Server ou autres plates-formes prises en charge*

* La réplication hétérogène (depuis Oracle ou vers un abonné non SQL Server) est une fonctionnalité dépréciée et sera supprimée dans une prochaine version !

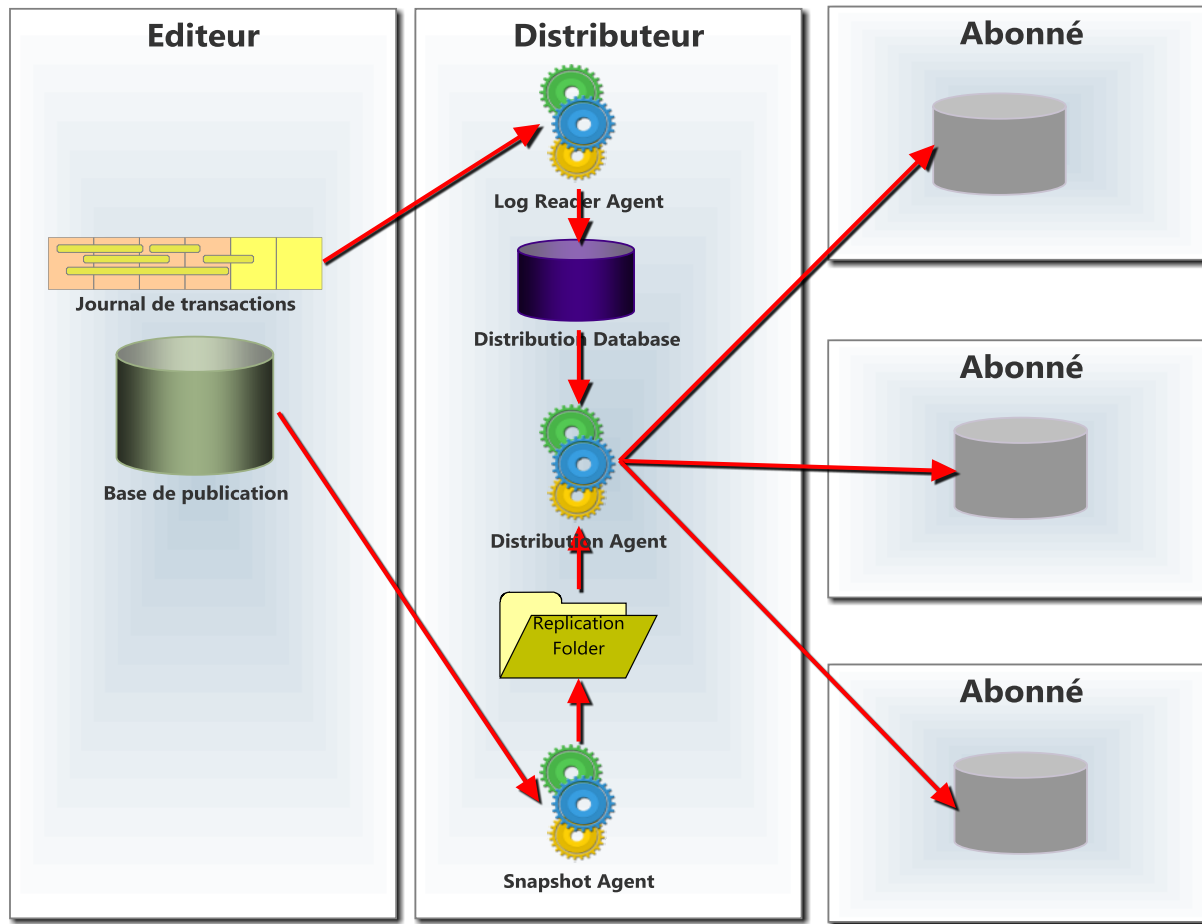


Ce qu'offre la réplication

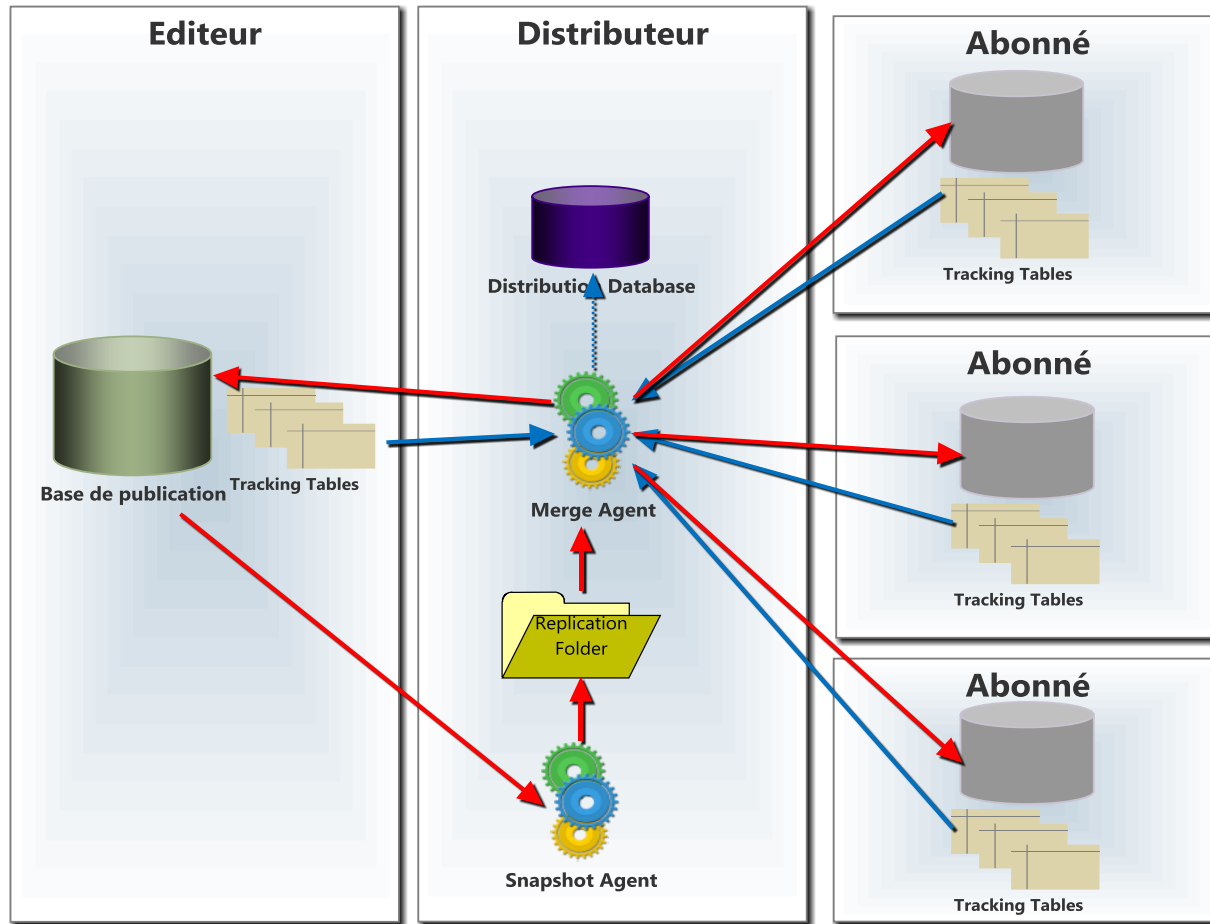
- 3 types de réplication
 - **Snapshot** : basée sur une copie des données (bcp) pour une première synchro ou un annule et remplace
 - **Transactionnelle** : basée sur un log reader, pour maintenir synchrones des réplicas essentiellement utilisés en lecture
 - **Fusion** (merge) : basée sur des triggers et des tables de tracking, pour des scénarios de synchro bi-directionnelle avec une prise en charge des conflits



Réplication transactionnelle



Réplication de fusion (merge)



Choix d'une solution de synchronisation

	Mise à jour sur les réplicas et gestion des conflits	Réplicas hétérogènes	Synchro au fil de l'eau	Edition Standard
Change Tracking	non	oui - via ETL	non	oui
Change Data Capture	non	oui - via ETL	non	non
Service Broker	non	non	oui	oui
Réplication snapshot	Updatable subscriptions	déprécié	non	oui
Réplication transactionnelle	Updatable subscriptions	déprécié	oui	sauf peer to peer
Réplication fusion	oui	non	oui	oui



Questions



<http://GUSS.pro/sqlsat>

...Et en plus on peut gagner des cadeaux



SQLSaturday Paris 2015

GUSS

