異種DB間対応
リアルタイムレプリケーション
DBMoto
(デービーモト)
～移行・連携・災害対策に～

株式会社クライム

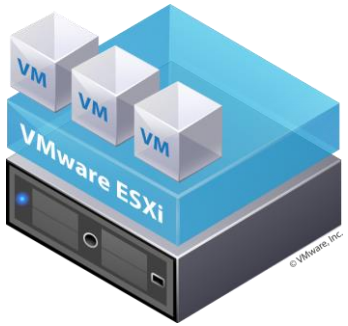
https://www.climb.co.jp/

レプリケーションとは

✓レプリケーション＝「複製」(≠ファイルコピー)

✓変更点(差分)のみを転送可能
⇒転送サイズ小、ネットワーク負荷小

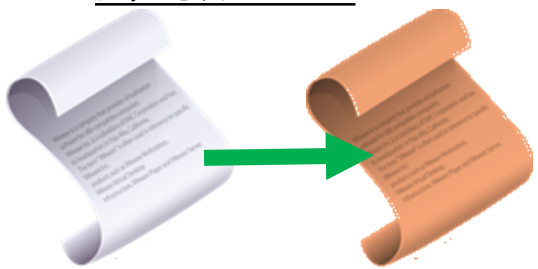
データベース



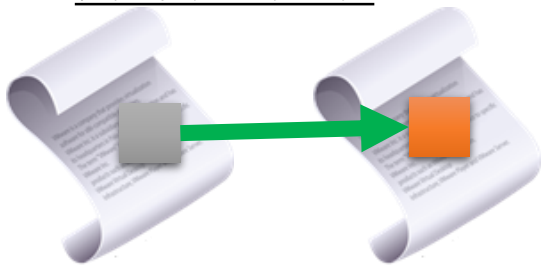
VMware/Hyper-V

NAS

ファイルコピー



レプリケーション

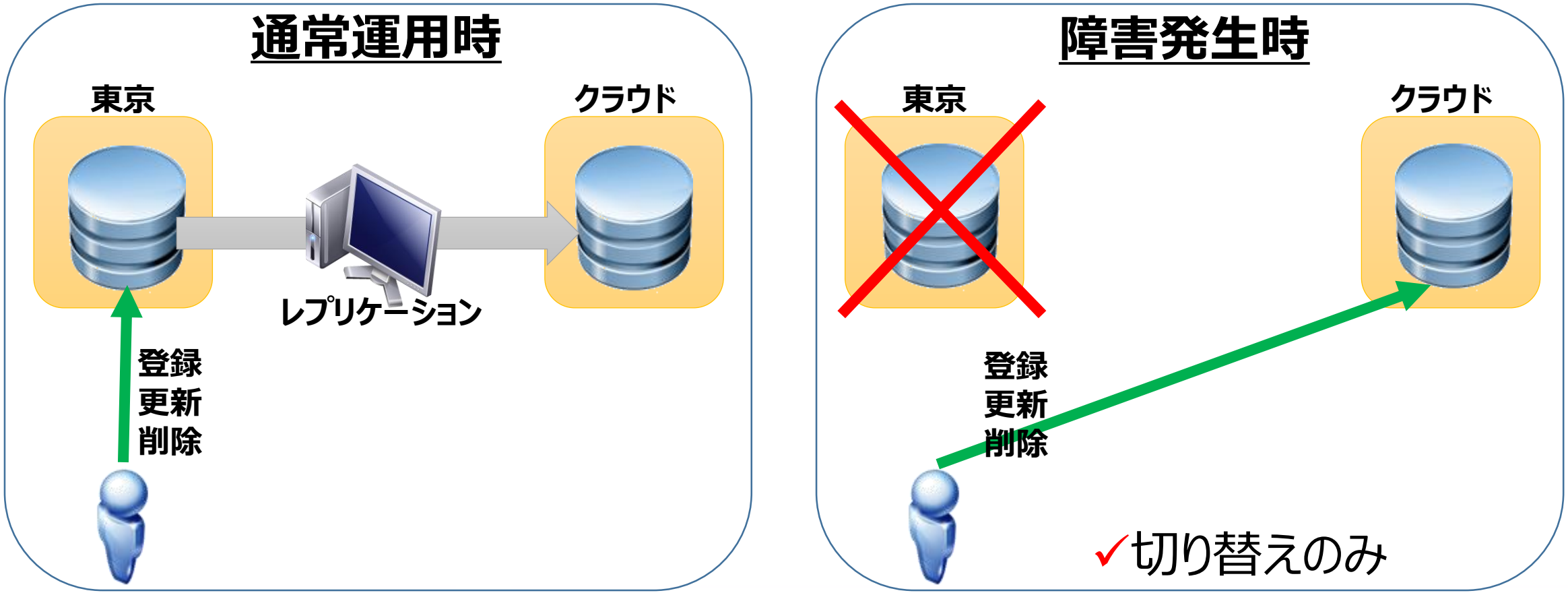


✓バックアップ
✓災害対策
✓データ連携
✓移行

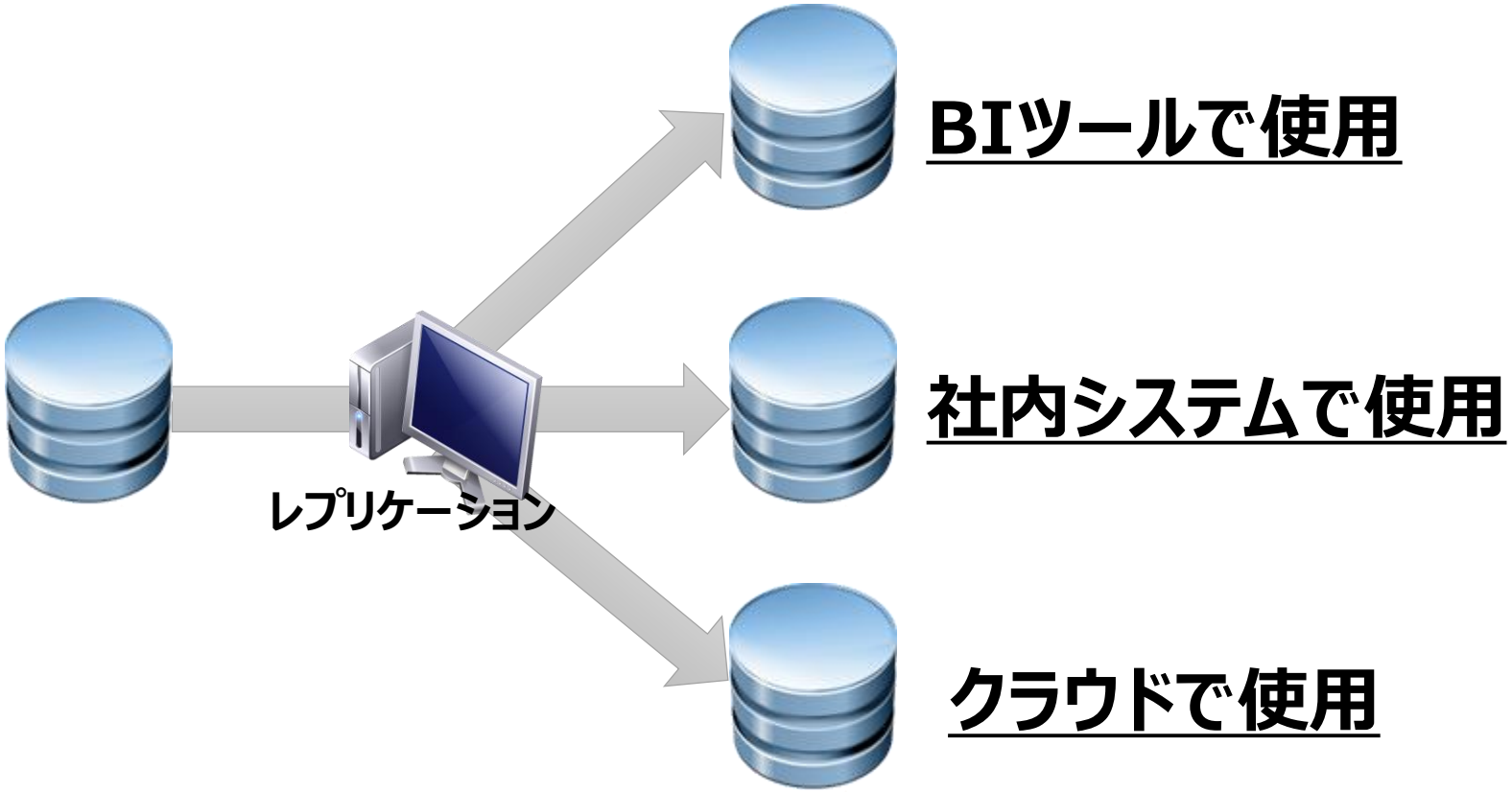
活用例: バックアップ

- ✓遠隔地でもクラウドでも
- ✓リアルタイム
- ✓自動化
- ✓低負荷
- ✓手間をかけず

活用例: 災害対策



活用例: データ連携



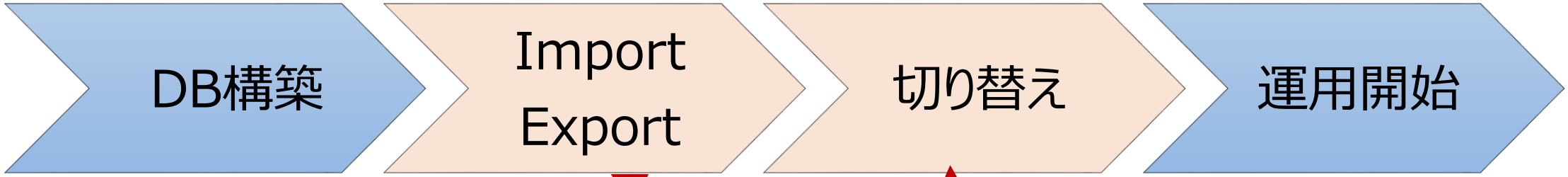
BIツールで使用

社内システムで使用

クラウドで使用

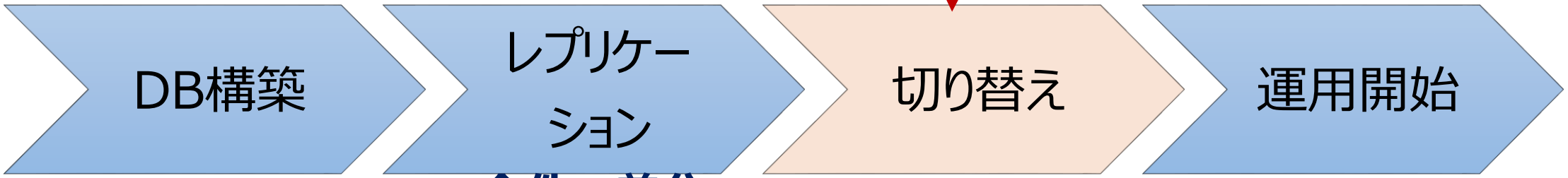
活用例：移行中にDBを止める必要なし

通常のDB移行



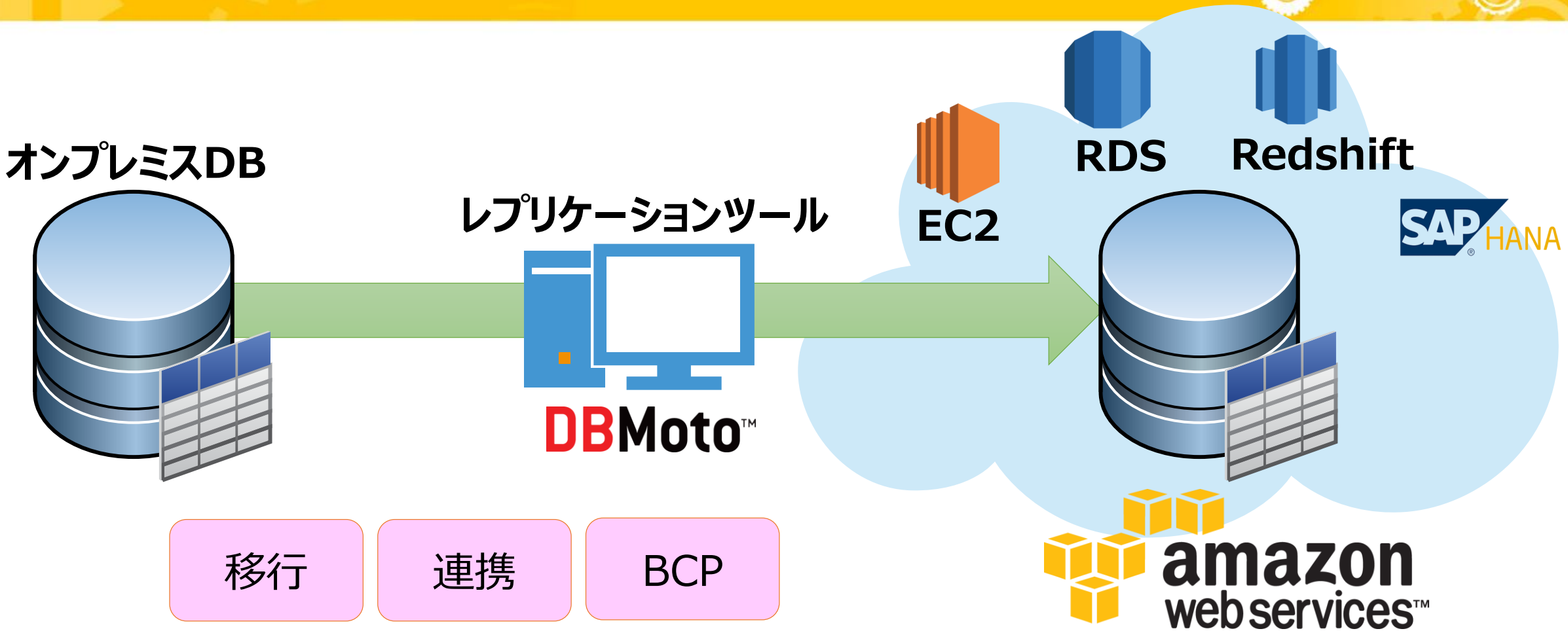
システム停止期間

レプリケーションによるDB移行



全件＋差分

活用例：オンプレミスとAWSを繋ぐ



AWS＋DBMoto：組合せは自由

オンプレ⇔EC2

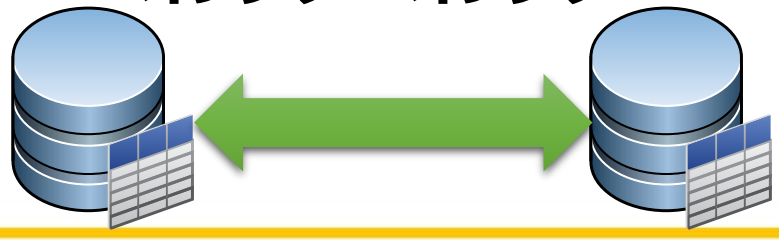
EC2⇔EC2

オンプレ⇔RDS

RDS⇔RDS

EC2⇔RDS

オンプレ⇔オンプレ



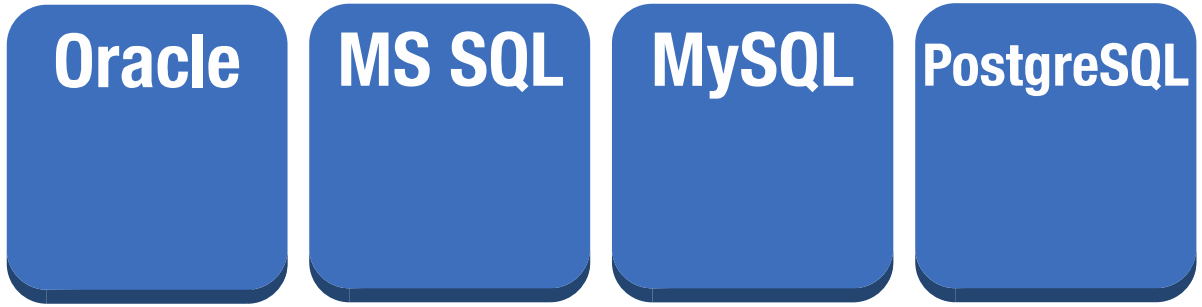
AWS+DBMoto: 対応サービス

Amazon EC2

- DBMotoがサポートする全てのDB
- Redshift/Aurora
- SAP HANA
- (DBMotoインストールマシン)

Amazon RDS

- Oracle
- MS SQL Server
- MySQL
- PostgreSQL



DBMotoの特徴

✓**対応力**
- 多くのDB・DWHをサポート
- DBはOS依存なし
- 物理、仮想、クラウド(AWS等)

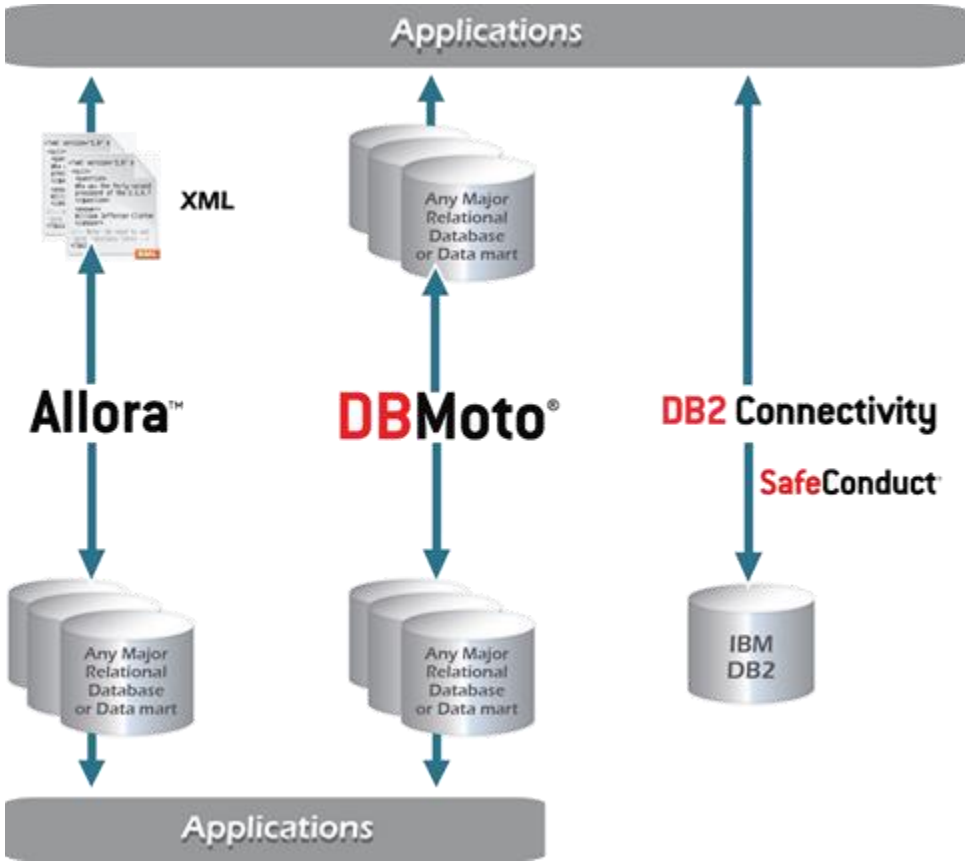
✓**使い易さ**
- Windowsによる一元管理
- エージェントレス
- ツールは日本語表示
- 導入簡単(1分以内)
- 設定簡単(コンサル不要)

✓**柔軟性**
- ほぼリアルタイム
- テーブル作成、スケジュール、関数、スクリプト
- 小規模から大規模環境まで

開発元：HiT Software, Inc.
- 1994年からDBアクセスツールの開発・販売
- 本社：サンノゼ, カリフォルニア
- DB2 expert ITリソース・パートナー
- 全世界に販売網(日本は株式会社クライムが担当)

http://www.hitsw.com/

HiT Software(開発元)製品ファミリー



DB2 Connectivity	DB2接続ドライバ ・HiT ODBC ・HiT JDBC ・HiT OLEDB ・Ritmo(.NET) ※DBMotoに同梱
DBMoto	リアルタイムDBレプリケーション
Allora	XMLとDB間の相互変換マッピング
SafeConduct	256ビットSSLデータ暗号化

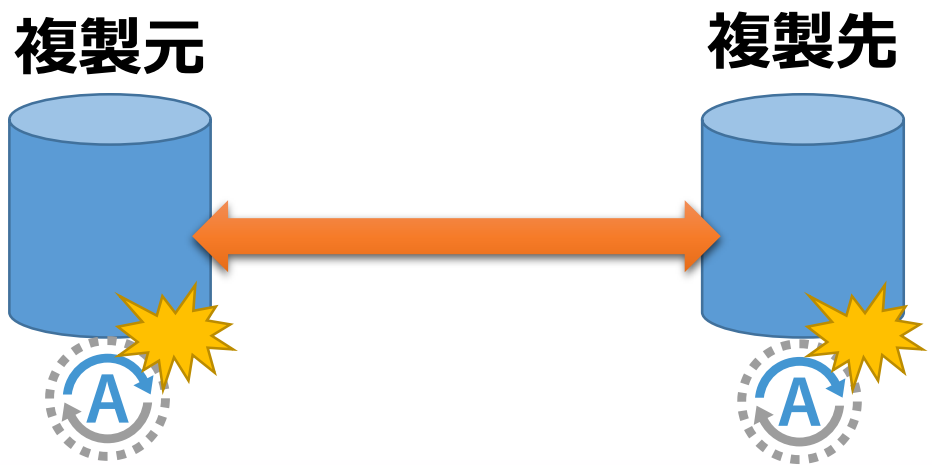
DBMotoと他社製品の位置づけ

DBMoto（エージェントレス）

- ◆Windowsマシンでの一元管理
- ◆エージェントレスによる低負荷

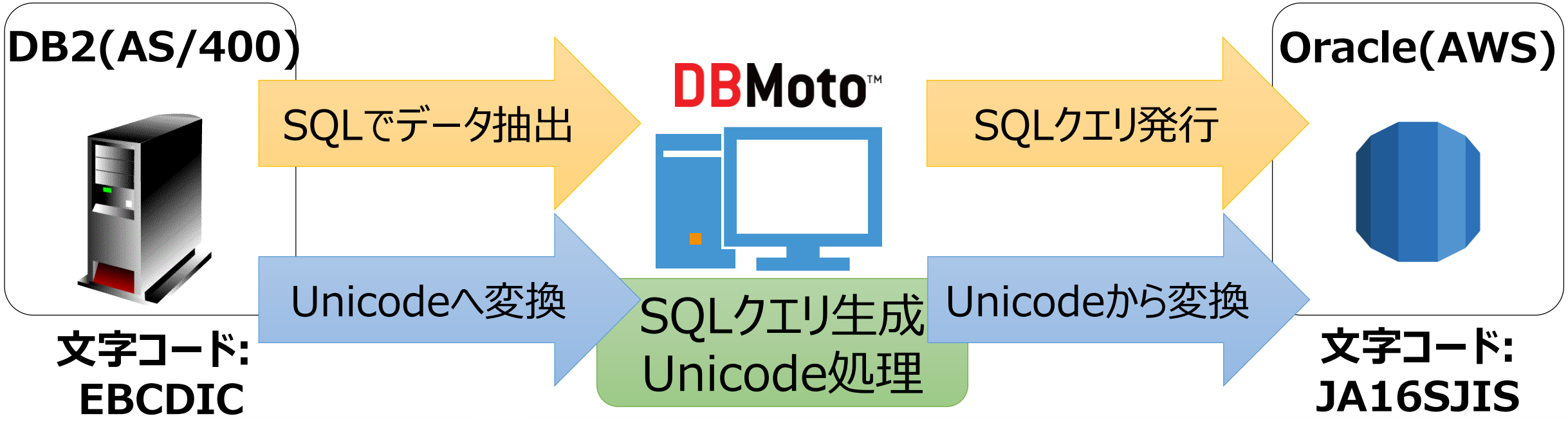
他社エージェントツール

- ◆即時性
- ◆ツールのOS依存なし



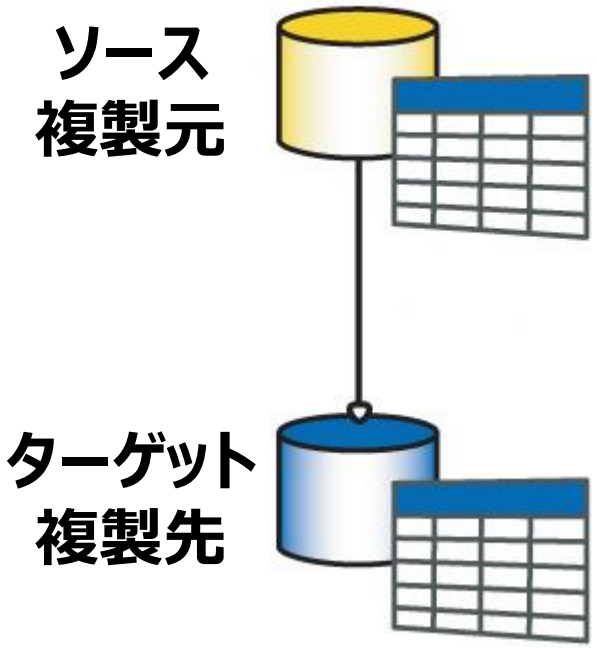
異種DB間対応、異種文字コード間対応

- 異種DB間対応：レプリケーションはすべて**SQLクエリ**で処理
- 異種文字コード間対応：文字コードはDBMotoにて**Unicode**で処理

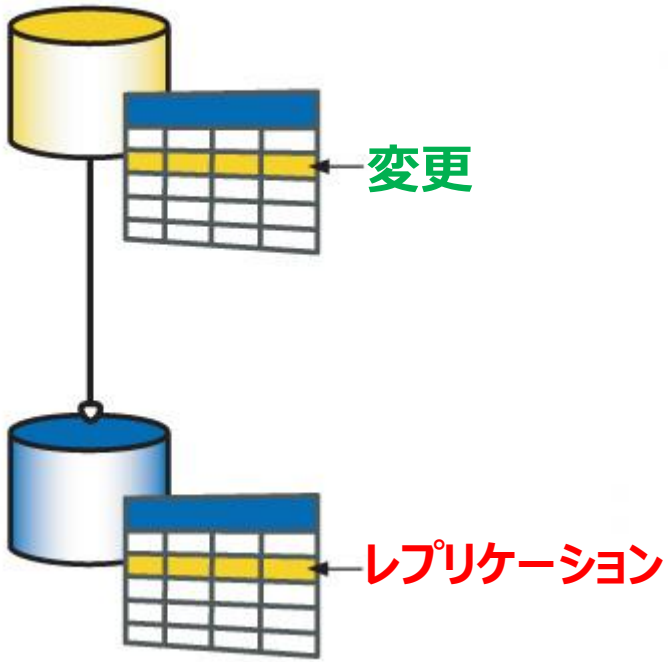


3つのレプリケーションモード

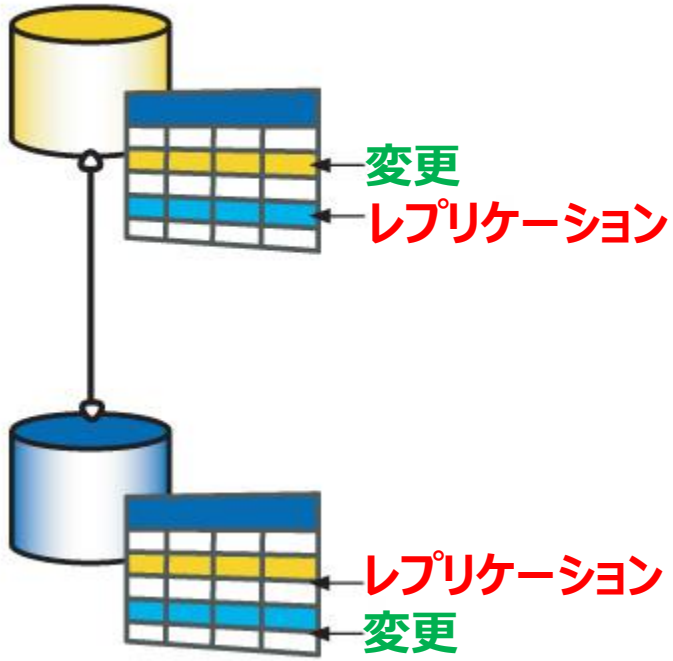
リフレッシュ
（全件）



ミラーリング
（片方向差分）

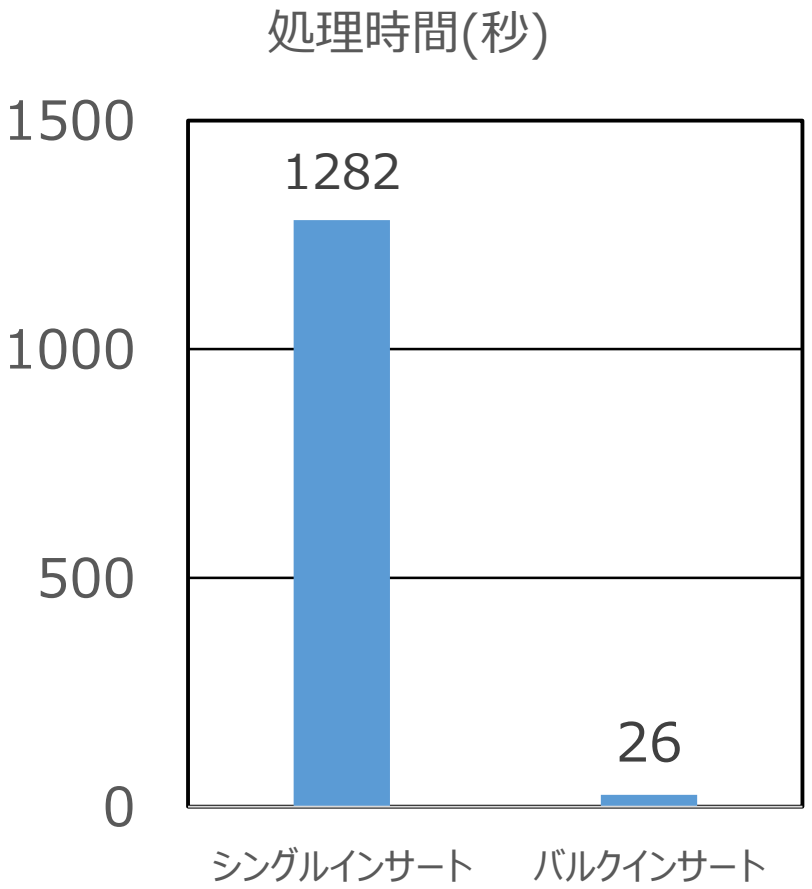


シンクロナイゼーション
（双方向差分）

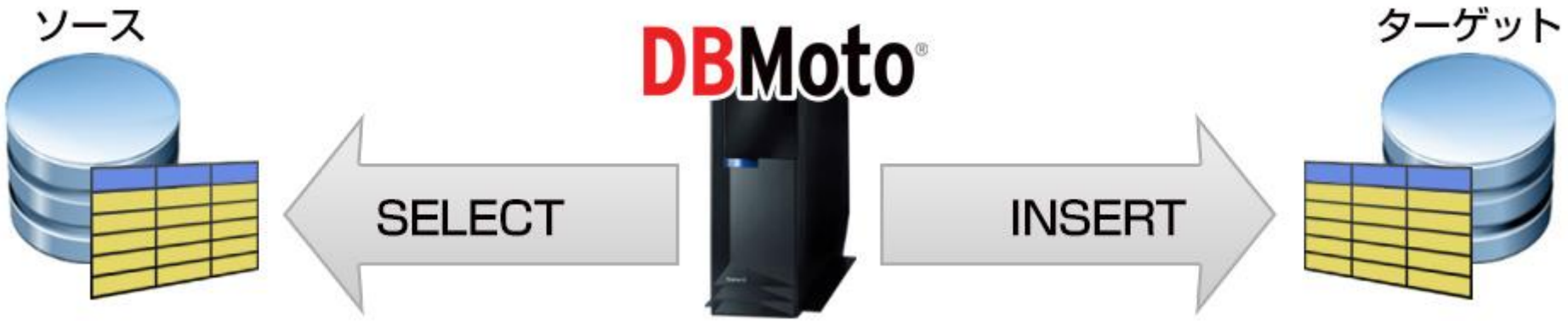


レプリケーションモード: リフレッシュ

- 選択したテーブルの全レコードを転送
- 初期レプリケーション
- スケジュールによる定期実行も可能
- バルクインサートによる高速転送



リフレッシュ：処理の流れ



参照先はテーブル

バルクインサートで高速転送

レプリケーションモード: ミラーリング

- 片方向の差分レプリケーション
- DBのトランザクションログを直接参照
- 差分のためデータ量小、負荷小
- DBにトリガーを設定することも可能
- 参照サイクルは既定で60秒(変更可能)

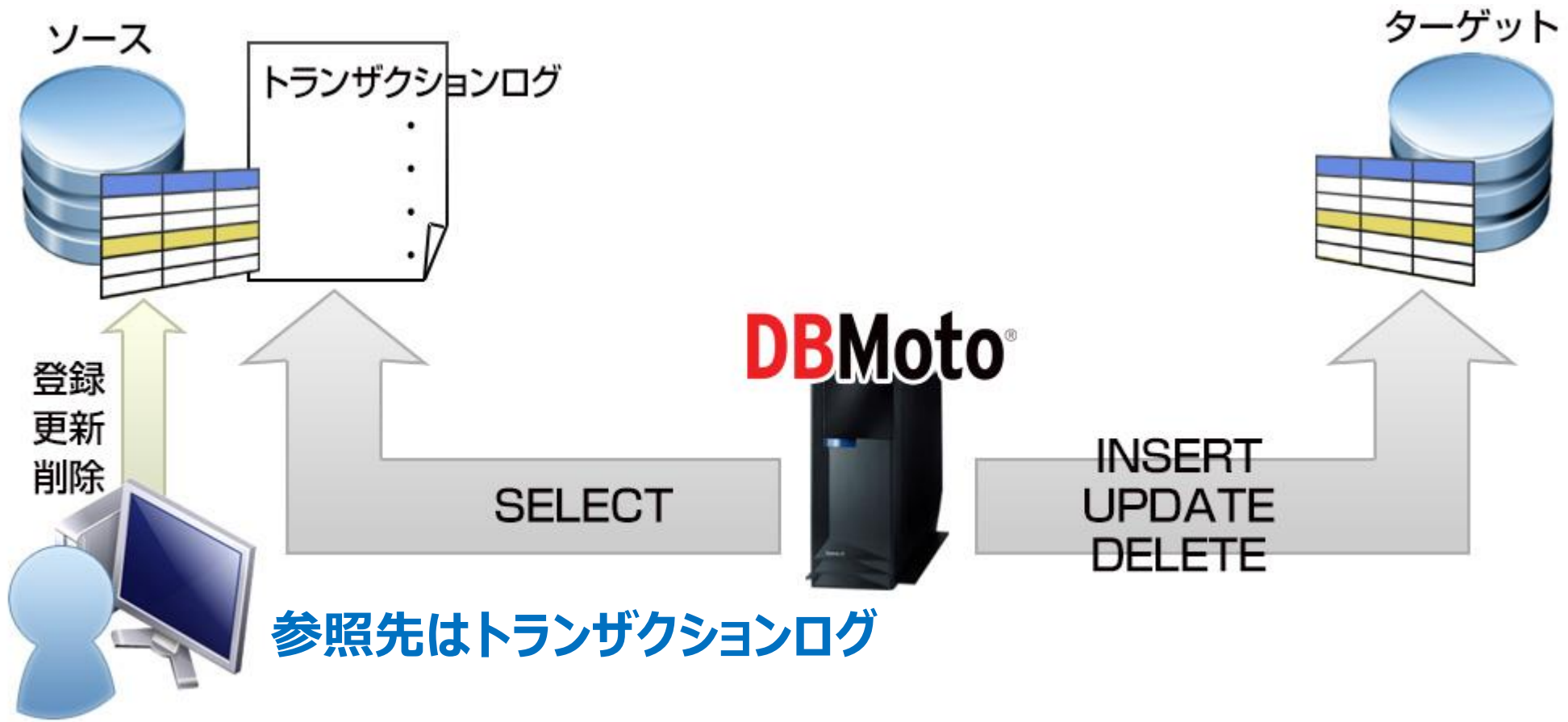
参照するトランザクションログ

DB2 AS/400	ジャーナル・レシーバー
DB2 LUW	ログ(プロシージャ経由)
Oracle	REDOログ・アーカイブログ
SQL Server	ログ(ディストリビュータ経由)
MySQL	バイナリログ

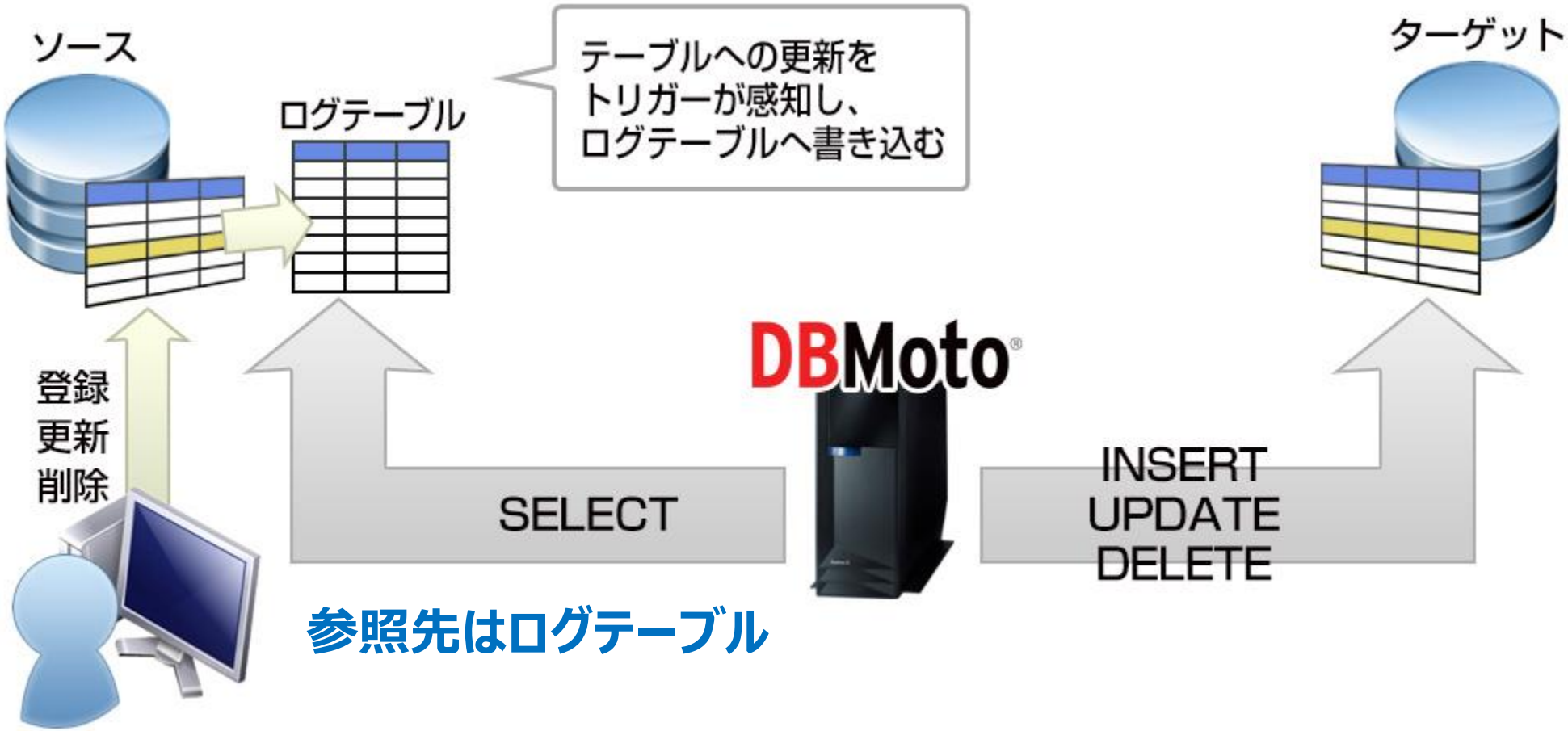
トリガー対応DB

・DB2 z/OS, DB2 LUW
・SQL Server
・MySQL
・Informix

ミラーリング：処理の流れ(トランザクションログ)



ミラーリング：処理の流れ(トリガー)



ミラーリング：整合性維持の仕組み

ソースDB: トランザクションログ

TID=1: Insert ～
TID=2: Update～
TID=3: Delete～
TID=4: Update～
TID=5: Update～

抽出

処理

ターゲットDB: 処理の流れ

Insert ～ (TID=1); Commit;
　DBMotoに**TID=1を保存**
Update～ (TID=2); Commit;
　DBMotoに**TID=2を保存**
Delete～ (TID=3); Commit;
　DBMotoに**TID=3を保存**
Update～ (TID=4); Commit;
　DBMotoに**TID=4を保存**
Update～ (TID=5); Commit;
　DBMotoに**TID=5を保存**

- 「クエリ発行⇒コミット⇒DBMotoにTID保存」を繰り返す
- 次の処理時はトランザクションログからTID＝6以降を参照する

ミラーリングの参照サイクル

・・・ 待機時間

・・・ レプリケーション処理時間

・ソースのトランザクションログを参照し、
ターゲットへレプリケーションするまでの一連の流れを含みます。
・処理するトランザクションレコード数によって時間は変動します。
・処理時間が参照サイクルを上回った場合は、
待機時間なしで次のレプリケーション処理に移ります。

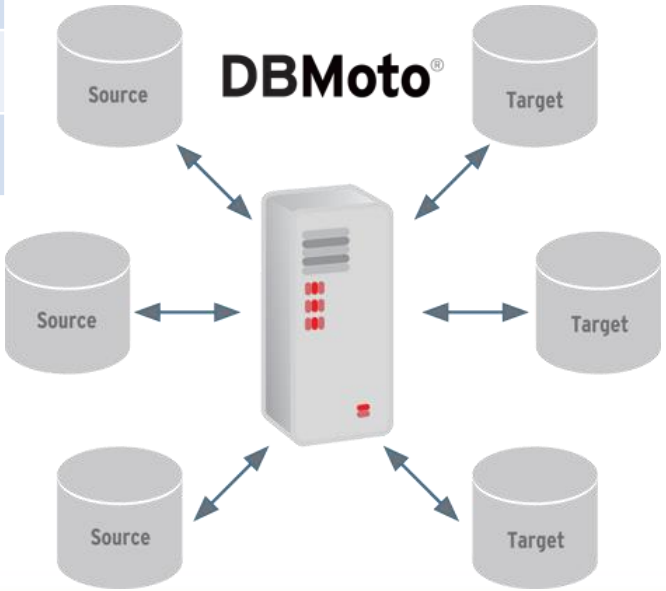
レプリケーションモード: シンクロナイゼーション

- 双方向の差分レプリケーション
- ミラーリングと手法は同等
- 双方のDBで更新が必要なアプリに
- 切り戻しも兼ねた災害対策に
- コンフリクト回避
- 3台以上のマルチシンクロナイゼーション

コンフリクト回避オプション

ソースを優先
ターゲットを優先
更新の早い方を優先
更新の遅い方を優先
ユーザスクリプト

マルチシンクロナイゼーション



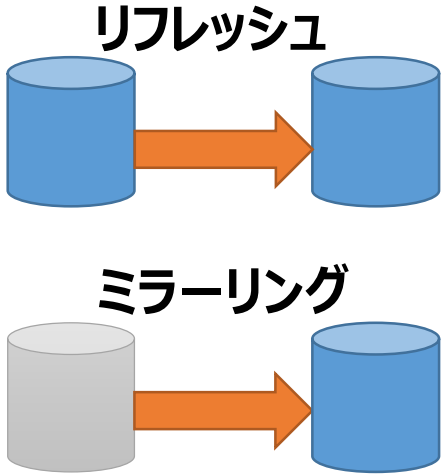
サポートするデータベース

リフレッシュ(全件)
ミラーリング、シンクロナイゼーション(差分)

IBM DB2 for i(AS/400)
IBM DB2 for z/OS
IBM DB2 for AIX, Linux, Windows
Oracle
MS SQL Server
MS Azure SQL Database
MySQL/Aurora
Gupta SQLBase
IBM Informix
SAP Sybase ASE
SAP Sybase SQL Anywhere
IBM PureData(Netezza)

リフレッシュ(全件)
ミラーリング(差分)のターゲット[複製先]

PostgreSQL/Redshift
SAP HANA
SAP Sybase IQ
Actian Vectorwise
HP Vertica
MS Access
Firebird
Ingres
IBM SolidDB



導入簡単・設定簡単

インストール

インストールは**1分以内**、
インストーラは軽量の**28MB**

設定

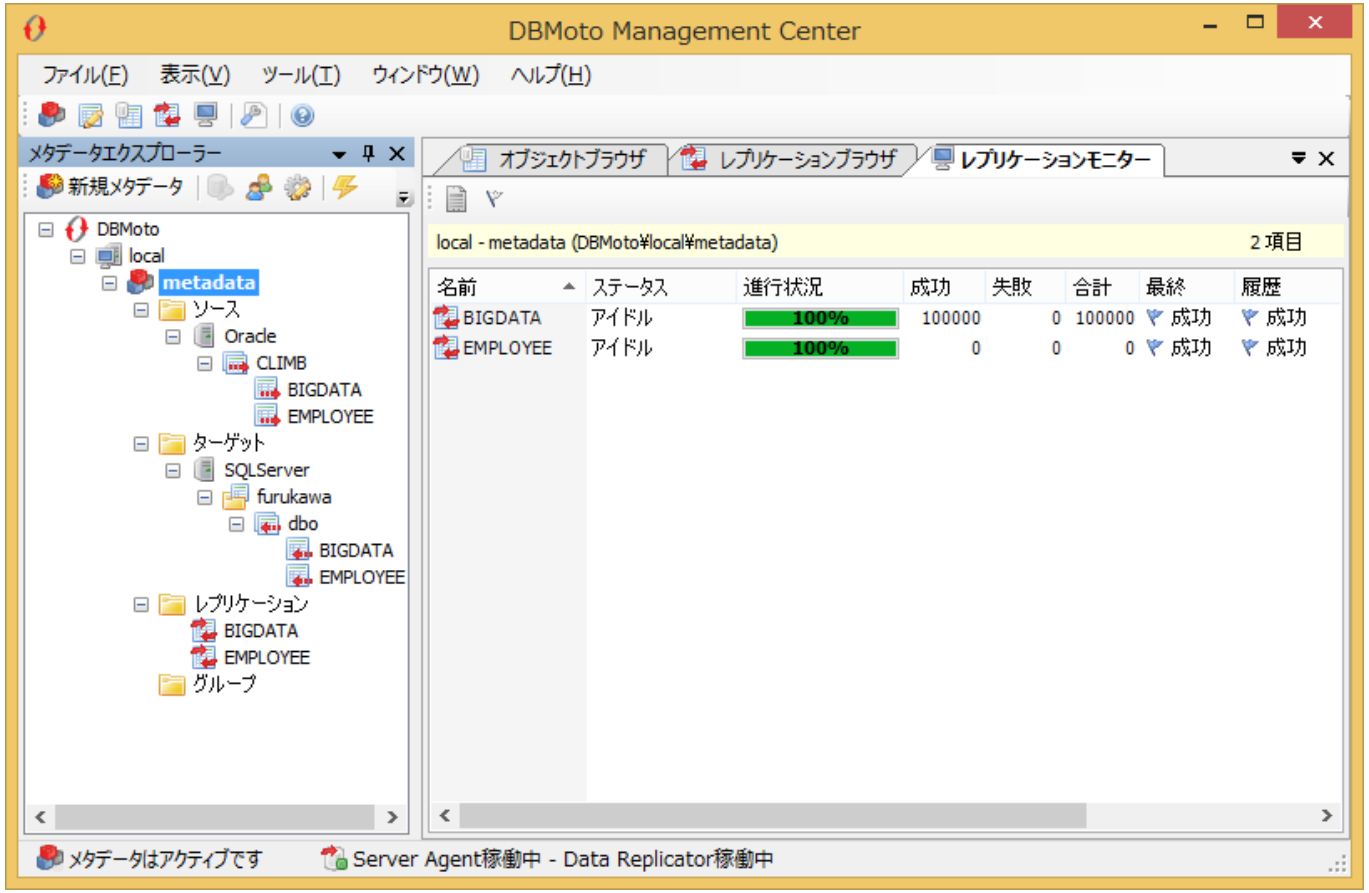
1. ソースDBへ接続
2. ターゲットDBへ接続
3. レプリケーションジョブを作成

処理

1. レプリケーションプロセス実行
2. 初期レプリケーション(リフレッシュ)
3. 差分レプリケーション(ミラーリング)

運用も簡単

- ◆日本語ローカライズ済み
- ◆レプリケーション進捗モニター
- ◆専用のログビューワ
- ◆結果比較・データ修復
- ◆メール通知(アラート)
- ◆DBMoto操作ユーザ権限
- ◆DBMoto冗長化対応
- ◆設定情報バックアップ・リストア



DBMotoの柔軟性

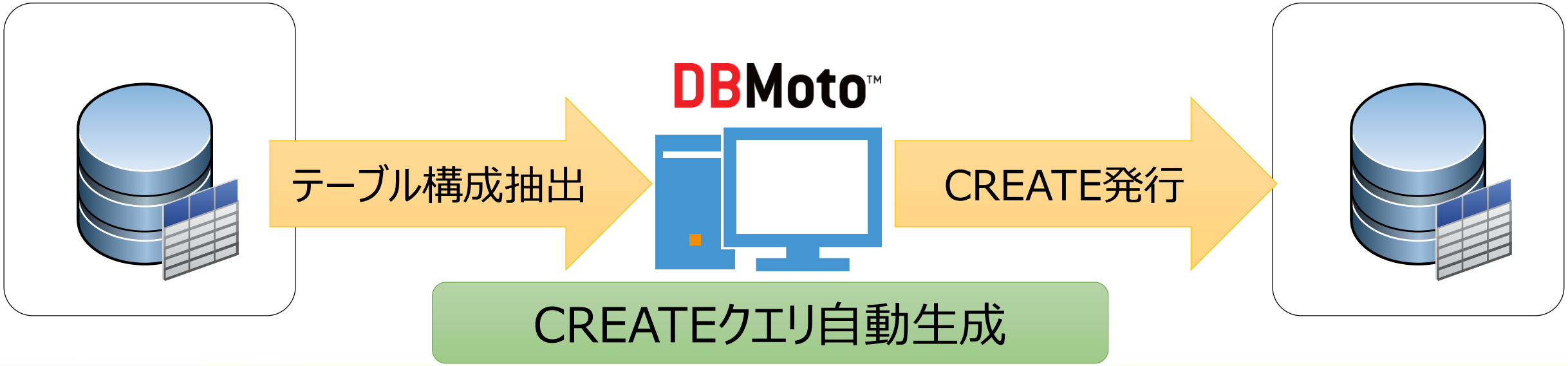
1. ターゲットDBにテーブルがない場合は？
2. PKがないとレプリケーションできない？
3. ジョブの作成単位は？
4. レプリケーションしてほしくない時間帯がある
5. テーブル構成・カラム数が異なる
6. レプリケーション時にデータ変換可能？
7. 条件付でレプリケーションしたい

DBMoto で簡単解決！

1. **DBMotoからテーブル作成**
2. **DBMotoの仮想PK機能**
3. **テーブル単位、一括作成機能あり**
4. **スケジュールでの除外設定**
5. **カラム単位でマッピング可能なため問題なし**
6. **関数を使用(VB, C#, ユーザ関数も可)**
7. **スクリプト記述で可能(VB, C#)**

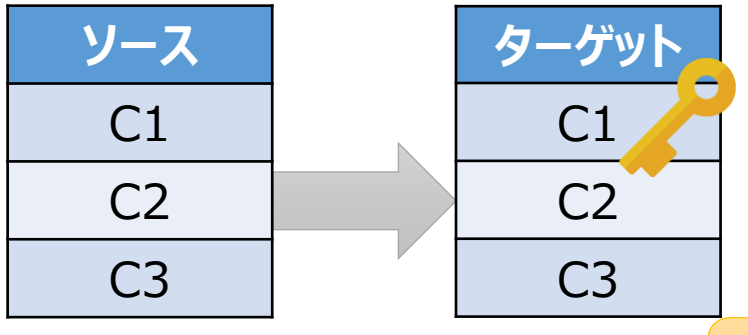
ターゲットへテーブル作成

- ソースのテーブル構成を元にDBMotoが自動でクエリを作成
- データタイプは自動で適切なものが選定される
- サイズ、PK、NOT NULLをそのまま引継ぐ
- 構成を手動変更することも可能



PKなしでも仮想PKでミラーリング可能

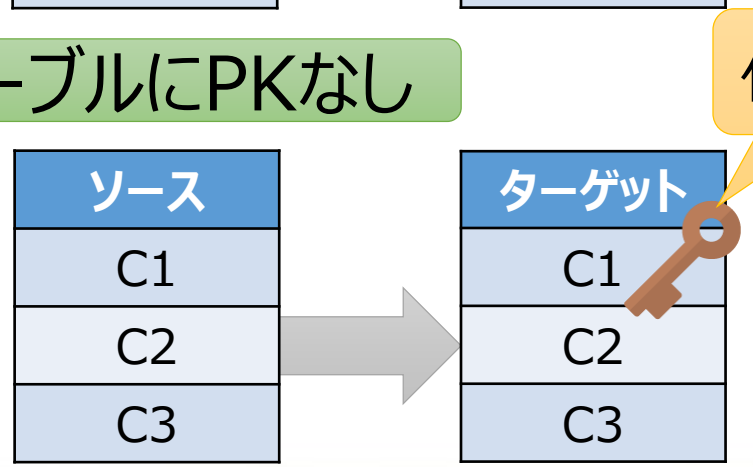
テーブルにPKあり



更新クエリ: update TABLE ～ where C1=xx

◆where句にPKが指定される

テーブルにPKなし



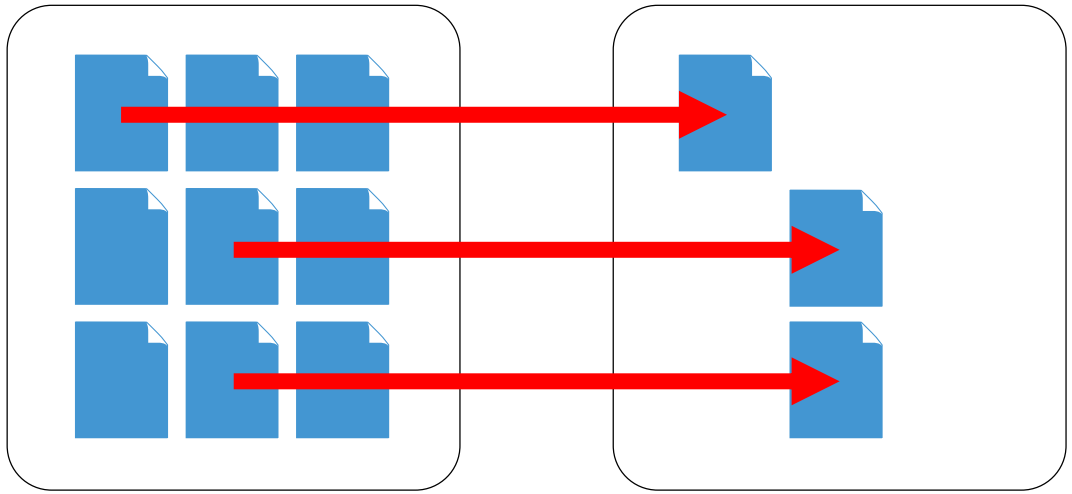
更新クエリ: update TABLE ～ **where C1=xx**

◆PKがないとwhere句が指定されない

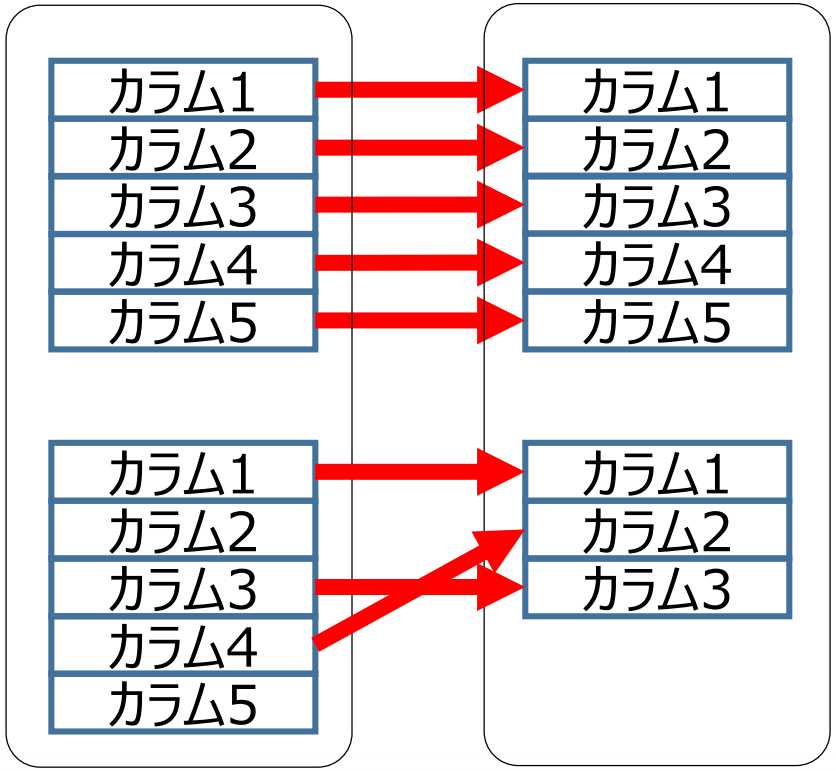
⇒仮想PKによってwhere句が指定される

必要なテーブルやカラムのみ連携

◆テーブル単位でジョブ作成
◆ジョブ一括作成も可能

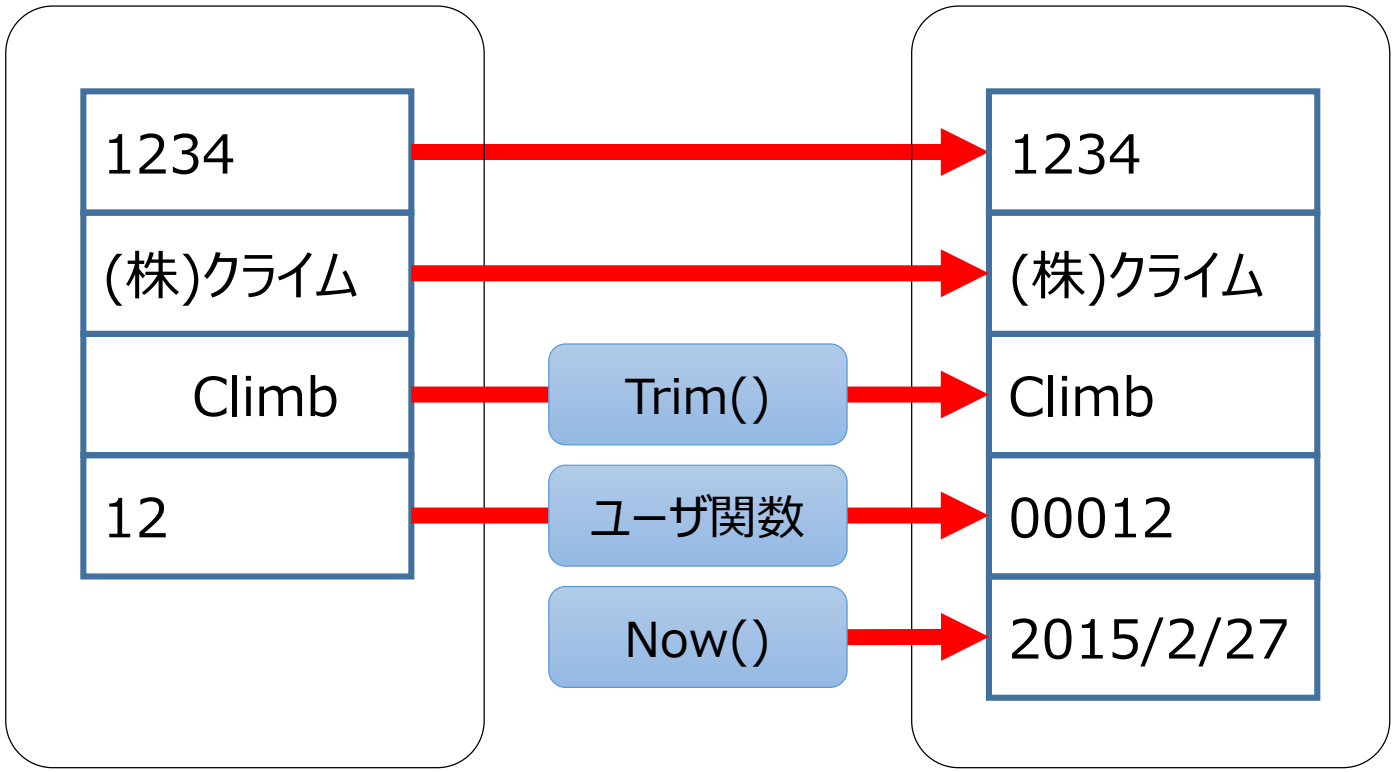


◆カラム単位でマッピング



データのカスタマイズ、複雑な連携

◆関数を使用可能: VB, C# 及びユーザ関数



◆スクリプト: VB, C#

・条件付きレプリケーション

例:値が～を満たすとき

例:更新のみ反映、登録は無視

・テーブルの結合

・ユーザ関数の定義

◆API: VB, C#, C++

・バッチ処理、パラメータ取得

導入事例：ローソンHMVエンタテイメント様

AS/400のデータをWindowsへ連携しDWHで活用

AS/400複数台を基幹システムで

DWH構築をSQL Serverで

DBMoto導入前：

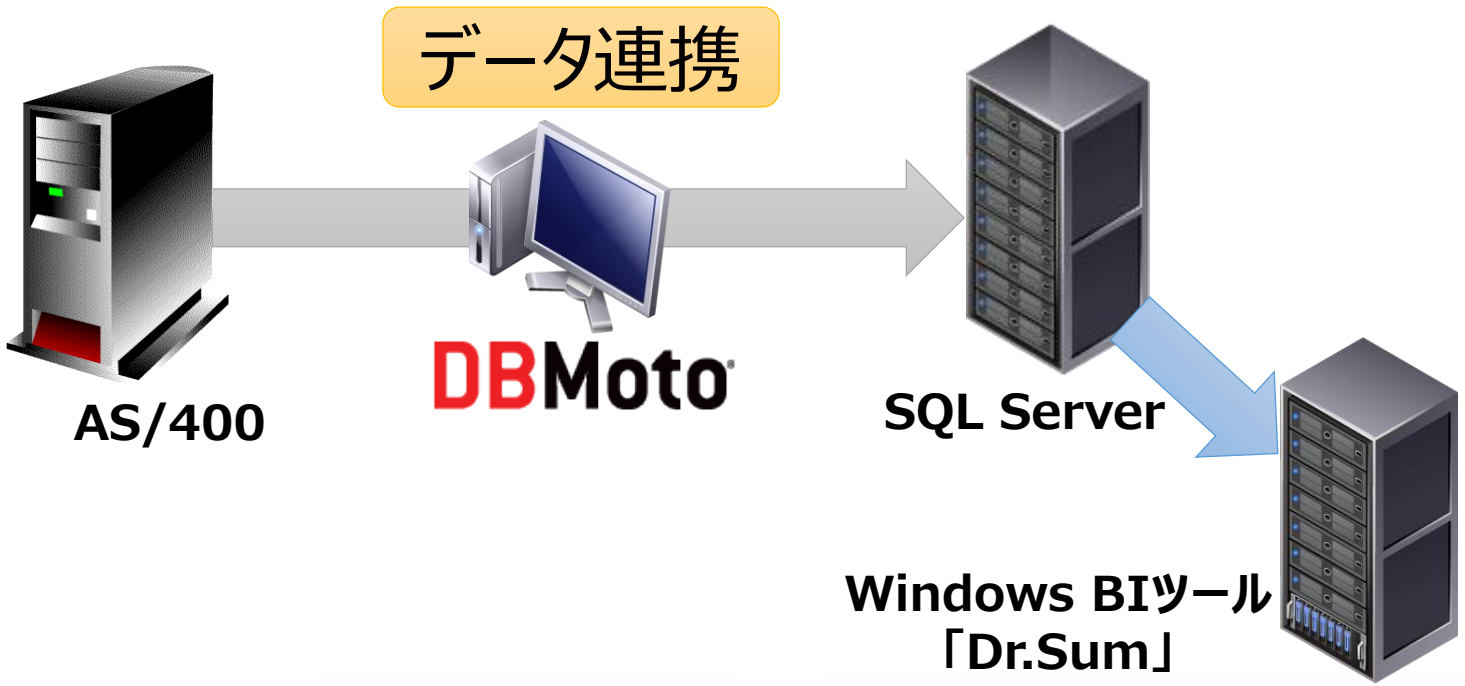
- ODBCによる手動での取り込み
- パフォーマンスが悪い
- リアルタイムの連携が困難

DBMoto導入後：

- ✓ データ統合による運用性の向上
- ✓ リアルタイムなデータ連携
- ✓ パフォーマンス向上

導入事例：クニミネ工業様

AS/400と連携できないWindowsツールの問題を解決



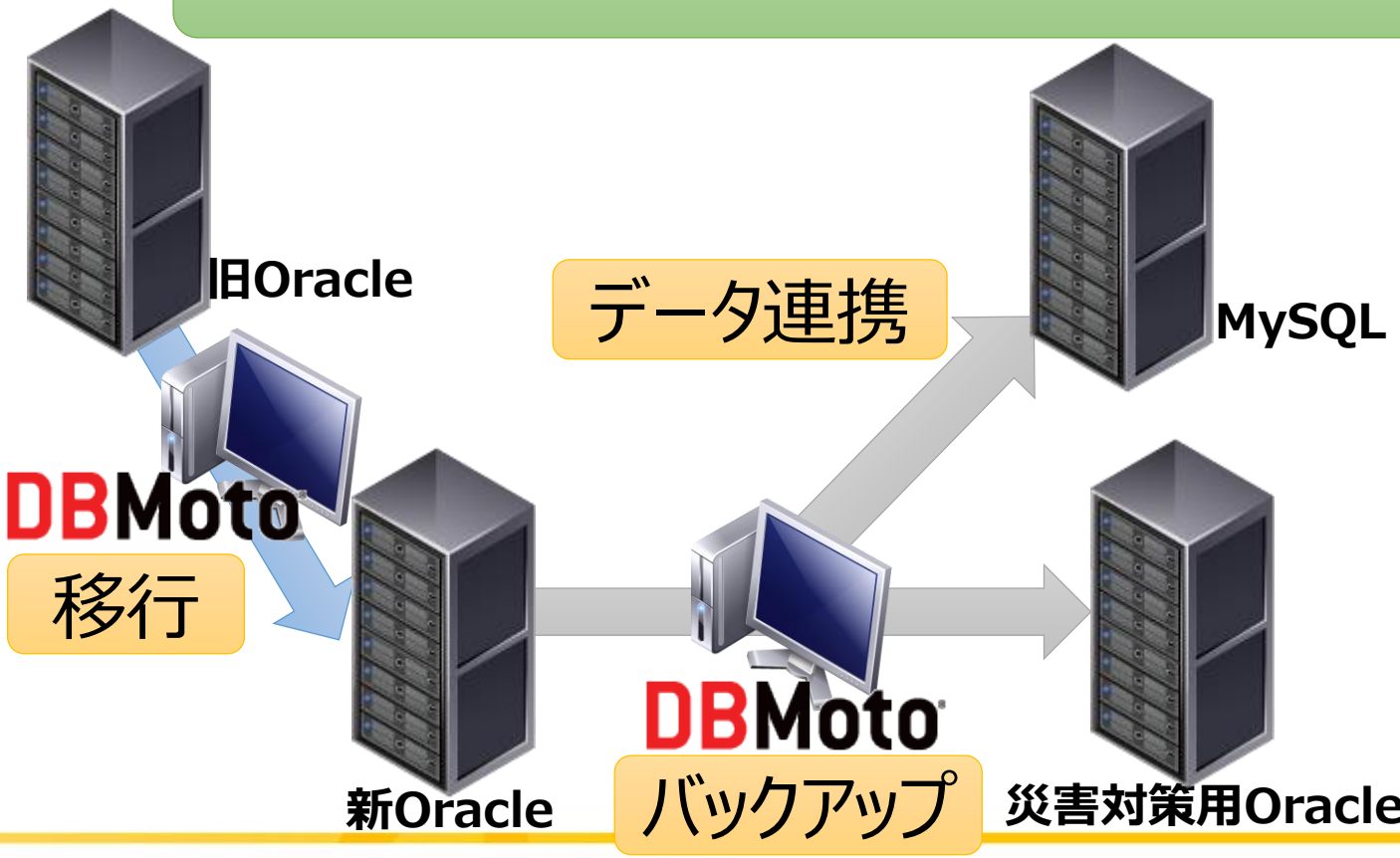
DBMoto導入前：
- AS/400とBIツールは直接連携不可
- データ抽出は手動

DBMoto導入後：
- ✓ AS/400⇒SQL Server⇒BIツール
- ✓ データ抽出はSQL Serverとの自動連携

32

導入事例：ぐるなび様

移行・バックアップ・データ連携の3パターン構成



DBMoto導入後：

＜移行＞

✓ システム停止時間を最小限に抑えてのOracleデータベースの移行

＜バックアップ＞

✓ ディザスタリカバリ環境構築により、障害時は切り替えのみでOK

＜データ連携＞

✓ 他システムへの連携用にMySQLへレプリケーション