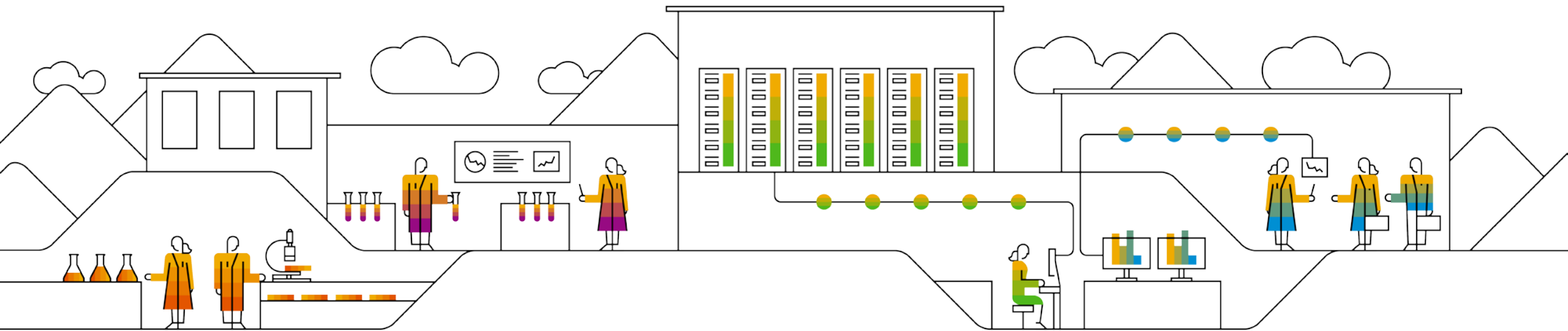




SAP SQL Anywhere and Mobile Link

2018年9月

PUBLIC

SQL Anywhere

スイートおよびサーバーについて



主な機能 : SQL Anywhere 17

DB機能

- 主要データタイプをフルサポート
- Transact-SQL と Watcom SQL の2種使用可
- 豊富なプログラミングインターフェース
- パラレル・クエリ（内部並列クエリ処理）
- OLAP系クエリー対応
- マテリアライズド・ビュー

可用性機能

- DBミラーリング機能(HA/DRクラスタ)
- 読み込み専用スケールアウト
- ログミラーリング機能
- ハードウェアリソース変動対応
- クラウド対応

運用系機能

- セルフチューニング ・ インメモリ・モード
- データベース全体/テーブルレベル/カラムレベル暗号化
- 通信暗号化 ・ カラムレベルのデータ圧縮
- 少リソース動作(最低20MBのメモリ)
- イベント(&ハンドリング)機能 ・ スケジューラー機能

SAP

SQL Anywhere 17



連携機能

- 統合 HTTP サーバー機能
- ODataプロデューサー
- リモートデータアクセス
- 外部認証（統合化認証）対応
- 外部環境プロシージャラー

含まれるコンポーネント



SQL Anywhereの主な機能の紹介

- データタイプをフルサポートしたDBMSです。
 - 標準タイプ
CHAR, LONG NVARCHAR, LONG VARCHAR, NCHAR, NTEXT, NVARCHAR, TEXT, UNIQUEIDENTIFIER, VARCHAR, BIGINT, BIT, DECIMAL, DOUBLE, FLOAT, INTEGER, NUMERIC, REAL, SMALLINT, TINYINT, MONEY, SMALLMONEY, LONG VARBIT, VARBIT, DATE, DATETIME, DATETIMEOFFSET, SMALLDATETIME, TIME, TIMESTAMP, TIMESTAMP WITH TIME ZONE, BINARY, IMAGE, LONG BINARY, UNIQUEIDENTIFIER, VARBINARY
 - 空間エンジンによる空間データタイプ
ST_Point, ST_LineString, ST_Polygon, ST_CircularString, ST_CompoundCurve, ST_CurvePolygon, ST_Geometry, ST_GeomCollection, ST_MultiPoint, ST_MultiPolygon, ST_MultiLineString, ST_MultiSurface
 - 全文検索：述語、ワイルドカード、類似検索を含む
 - OPENXML を介した XPATH クエリーによる XML タイプ
- 2つのSQL
 - 方言としては Transact-SQL と Watcom SQL の2種が使用可能
 - 1つのDB内で混在可能です。
 - プロシジャAはTransact SQL、プロシジャBはWATCOMということはOK
 - プロシジャAのコードの前半はTransact、後半がWATCOMはNG
 - SQL:2008準拠（1992,1999,2003）



提供プログラミングインターフェース

SAP SQL Anywhere では下記のアプリケーションプログラミングインタフェースを提供しています。

- **ADO.NET**
 - Microsoft .NET Framework バージョン 2.0、3.0、3.5、4.0、4.5に対応
 - Entity Framework 5.0 および 6.0対応 (Visual Studio 2015対応)
- **JDBC**
 - Type2とType4の2種類提供
- **ODBC**
- **OLE DB / ADO**
- **Embedded SQL**
- **SQL Anywhere データベース API for C/C++**
- **Open Client**
- **OData**
- **XS JavaScript**
- **Node.js**
- **Perl DBI**
- **Python**
- **Ruby**

SQL Anywhereの主な機能

- 動的パラメータ管理、セルフチューニング

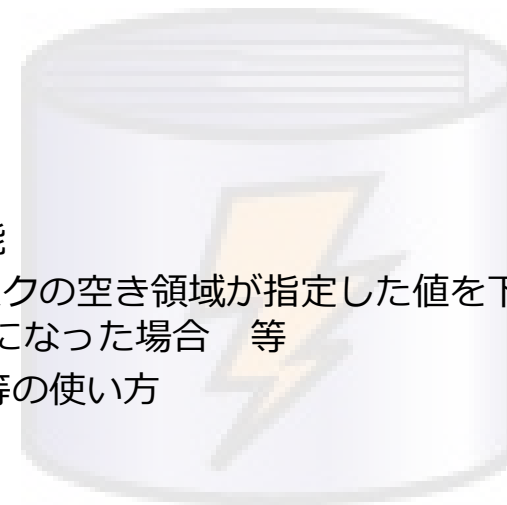
- キャッシュサイズ、同時実行タスク数等リソース関連のパラメータは自動調整を行うものが殆どです。
- 動作中のOSのメモリ使用状況を監視し、ユーザー要求処理時にメモリに余裕があるようであればキャッシュを追加、ユーザー処理要求が少ない状態でOSのメモリ使用状況が逼迫した状態であれば余剰を開放
- データベースのスループットを監視し、同時実行処理数を最も効率の良いスループットになるよう自動的に変動。（山登り法と放物線近似法を組み合わせで自動調整）

- スケジュール機能

- DBMS自体が時間指定で決められたコマンド・プロシジャを実行するスケジューリング機能（別スケジューラー不要）
- バックアップ等の自動化
- バッチ処理の自動化、月次処理で集計テーブルを作成する等

- イベント機能（&ハンドリング機能）

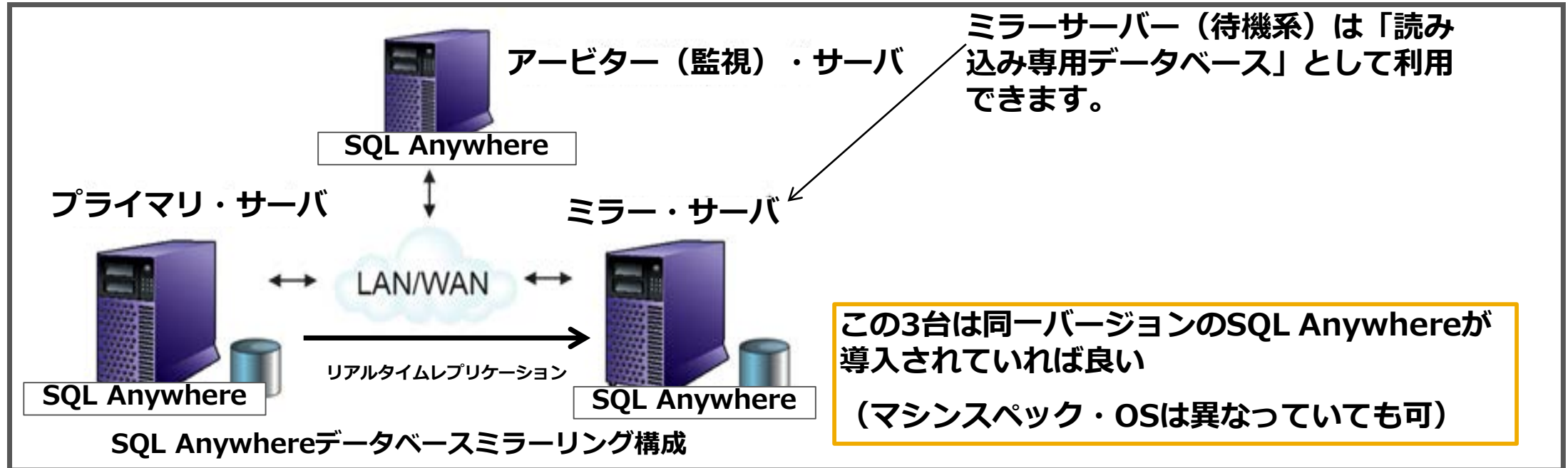
- 予め定めたイベントが発生した場合に決められたコマンド・プロシジャを実行する機能
- DBが起動・終了する際、ユーザーが接続・切断した際、DBファイルを配置したディスクの空き領域が指定した値を下回った際、DBのファイルサイズが指定したサイズに達した場合、指定した時間以上DBがアイドル状態になった場合 等
- アイドル状態になったらテーブルの断片化をチェックして必要であれば再編成 等の使い方



SQL Anywhereの主な機能

- データベースミラーリング機能

SQL Anywhereの機能による高可用性クラスタ・DRクラスタ構築機能です。**特別なハードウェアや他のソフトウェア、クラスタリングソフトウェアは必要ありません。**主系・待機系としてDBが動作するマシン2台とそれら2台を監視する小さなプロセスが動作するマシンの計3台で動作します。



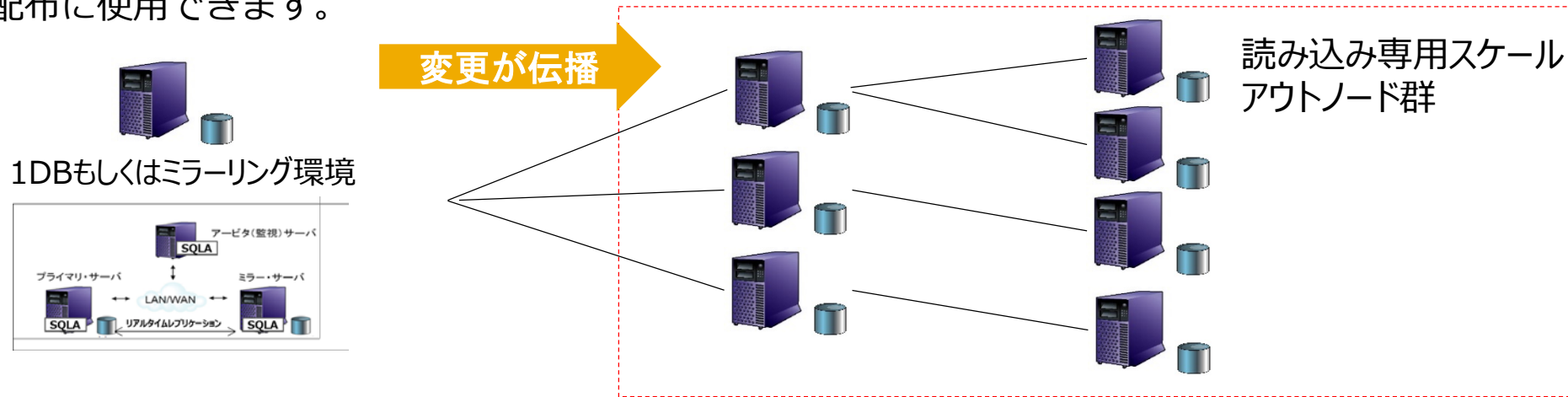
プライマリ→ミラー間はトランザクションログ送信によるリアルタイムレプリケーションです。

- ログの送信・反映タイミング違いの3モード

SQL Anywhereの主な機能

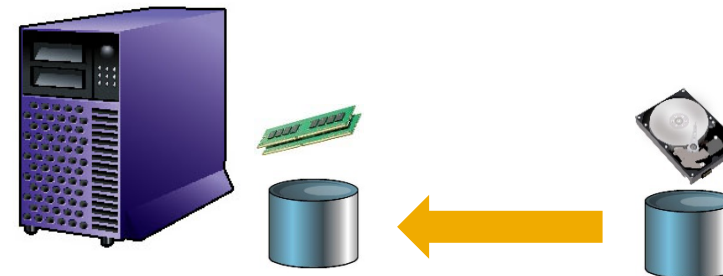
- 読み込み専用スケールアウト

- データベースを複製したノードを複数作成し、レポート作成やデータ参照など多数の要求に対する負荷軽減やデータ配布に使用できます。



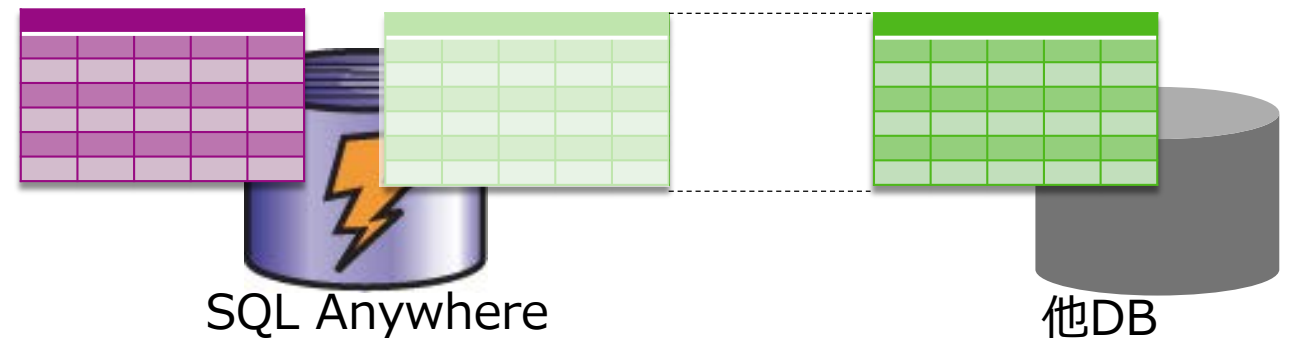
- インメモリ・モード

- データベースファイルをメモリ上に展開することによりI/Oを高速に
- 書き込み系処理に効果を発揮
- データの保全レベルを選択可
 - 完全インメモリ
 - 非同期遅延ディスク書き込み



SQL Anywhereの主な機能

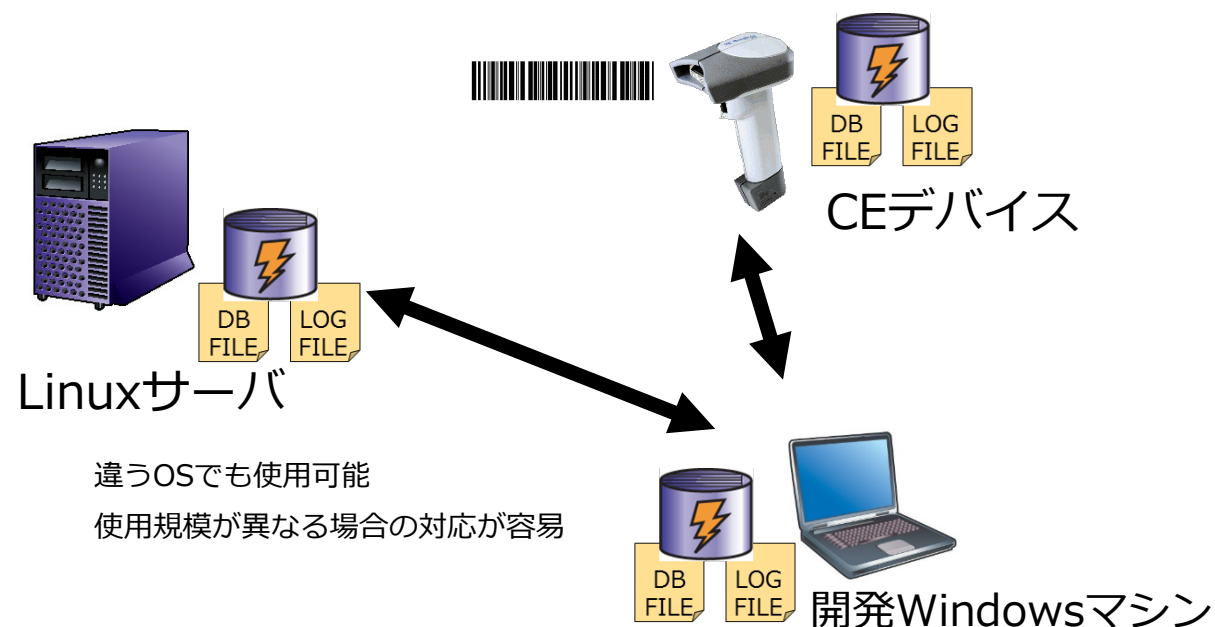
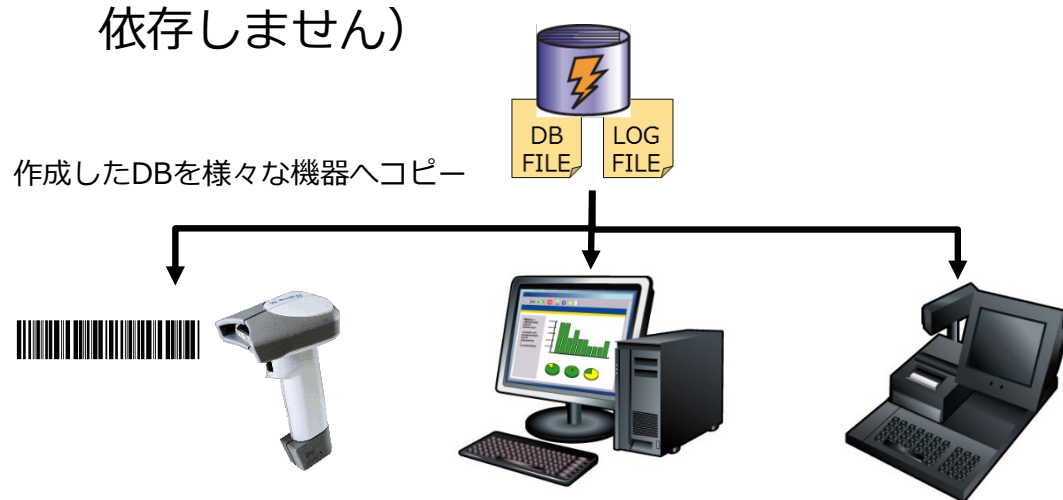
- 統合 HTTP サーバー機能
 - SAP SQL Anywhere サーバー自体をHTTPサーバとして利用可
 - プロシジャでWebアプリケーションを組むことが可能
 - HTTPサーバのURLに返り値としてHTMLを返すプロシジャを割当て→アクセスで実行
 - **小型Web アプリケーションの自己完結型のデプロイメント**
- SQL Anywhere単体でOData Producerに。
- リモートデータアクセス
 - リンクテーブル、データベースリンク、 Smart Data Access…等と呼ばれる他DBのテーブルを自DBにあるかのように見せる機能
 - SQL Anywhere、 ASE、 IQ、 SAP HANA
 - IBM DB2 / Microsoft Access / Microsoft SQL Server
 - MySQL/Oracle Database /その他ODBC



データベースファイル

SQL Anywhereのデータベースを構成するファイルは「データファイル」と「ログファイル」の2種

- データファイル：データを格納、ログファイル：トランザクションログを格納
- データファイルは最大13個
 - 最大1ファイル32KBページサイズ設定で8TB（or OS / ファイルシステムの最大ファイルサイズ）
*13個
- ファイルコピーでの配布が可能です。
- **DBを作成したOSでないOSへのコピーも可能です。**（DBファイルとログファイルはプラットフォームに依存しません）



SQL Anywhere サポートプラットフォーム

Windows

Windows 7 x86 and x64

Windows 8.x x86 and x64

Windows 10 x86 and x64

Windows Server 2003 x86 and x64

Windows Server 2008 x86 and x64

Windows Server 2008 R2 x64

Windows Server 2012 x64

Windows Server 2012 R2 x64

Windows Server 2016

Windows Mobile for Pocket PC and Smartphone

Windows Mobile 6.x Classic, Professional and Standard (ARMv4i) ver16

Windows CE 5.x and 6.x (ARMv4i) ver16

Linux for x86 and x64

Red Hat Enterprise Linux 5/6/7

SUSE Linux Enterprise 11/12

Ubuntu 12/14/16

Linux for ARM

Raspberry Pi on Debian GNU/Linux

Olimex A20 on Debian GNU/Linux

Solaris

10 SPARC and x64

11 SPARC and x64

HP-UX

11i v2 IA64 on Itanium 2

11i v3 IA64 on Itanium 2

AIX

6.1 POWER

7.1 POWER

Mac OS X

10.9 x64

10.10 x64

詳細は

<http://scn.sap.com/docs/DOC-35654>

を御参照下さい

参考) Ultra Light

Windows

Windows PC版はSQL Anywhereに準ずる

Windows Mobile 5 for Pocket PC and Smartphone
(Windows CE 5.0) ARMv4i

Windows Mobile 6.x Classic, Professional and Standard
(Windows CE 5.x) ARMv4i

Windows CE 6.0 ARMv4i

Windows Phone 8.1 ARM/x86(for emulator only)

Universal Windows Platform apps (UWP) ARM, x86, x64
(use WinRT components)

Android

4.1 or later ARM v7,x86, x86_64

iOS

Current version

Linux

SQL Anywhere に準ずる (x64)

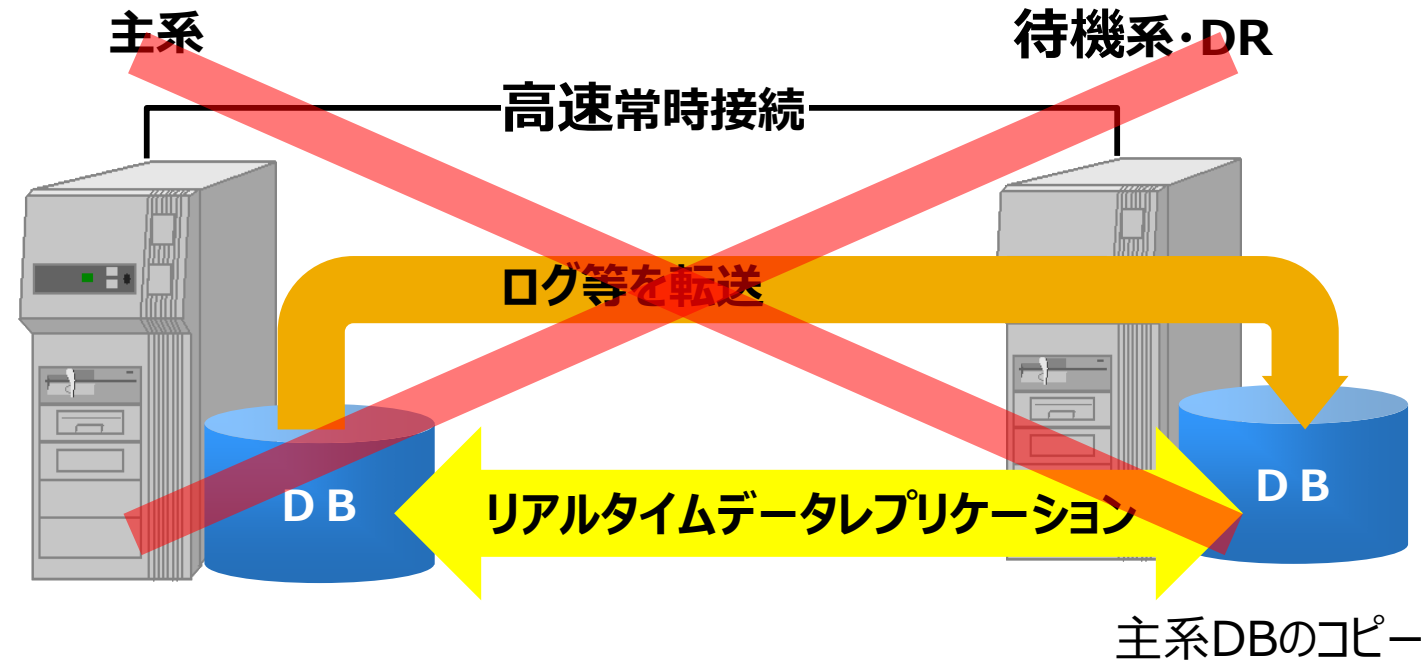
Mac OS X

10.9+ x86_64

SQL Anywhere の Mobile Link について

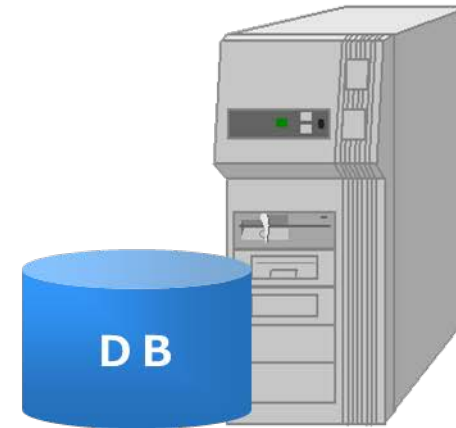


Mobile Linkとは？



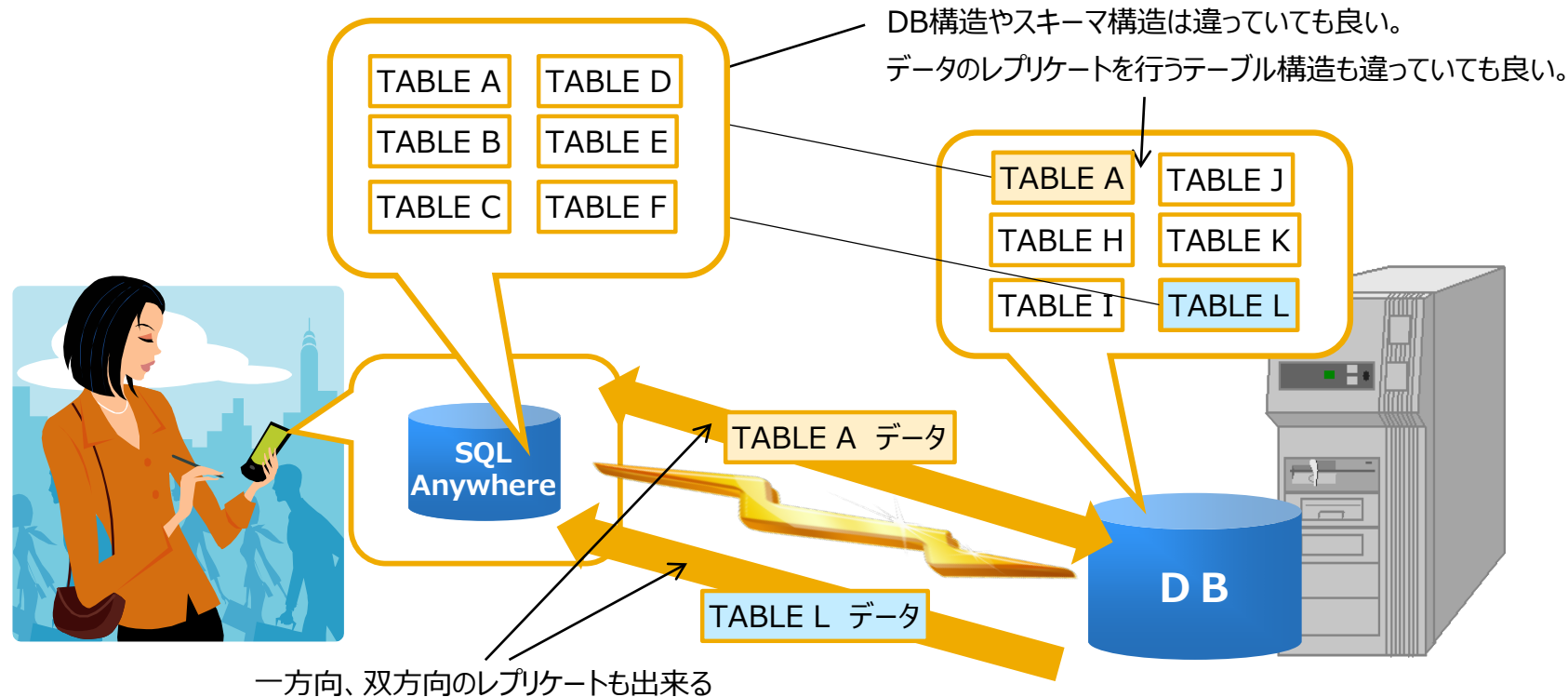
こういうものではありません！

Mobile Linkとは？



- **常時接続を必要としないデータレプリケーションソリューション**
⇔リアルタイムレプリケーションではない
→データ同期を実行する時のみ接続する。

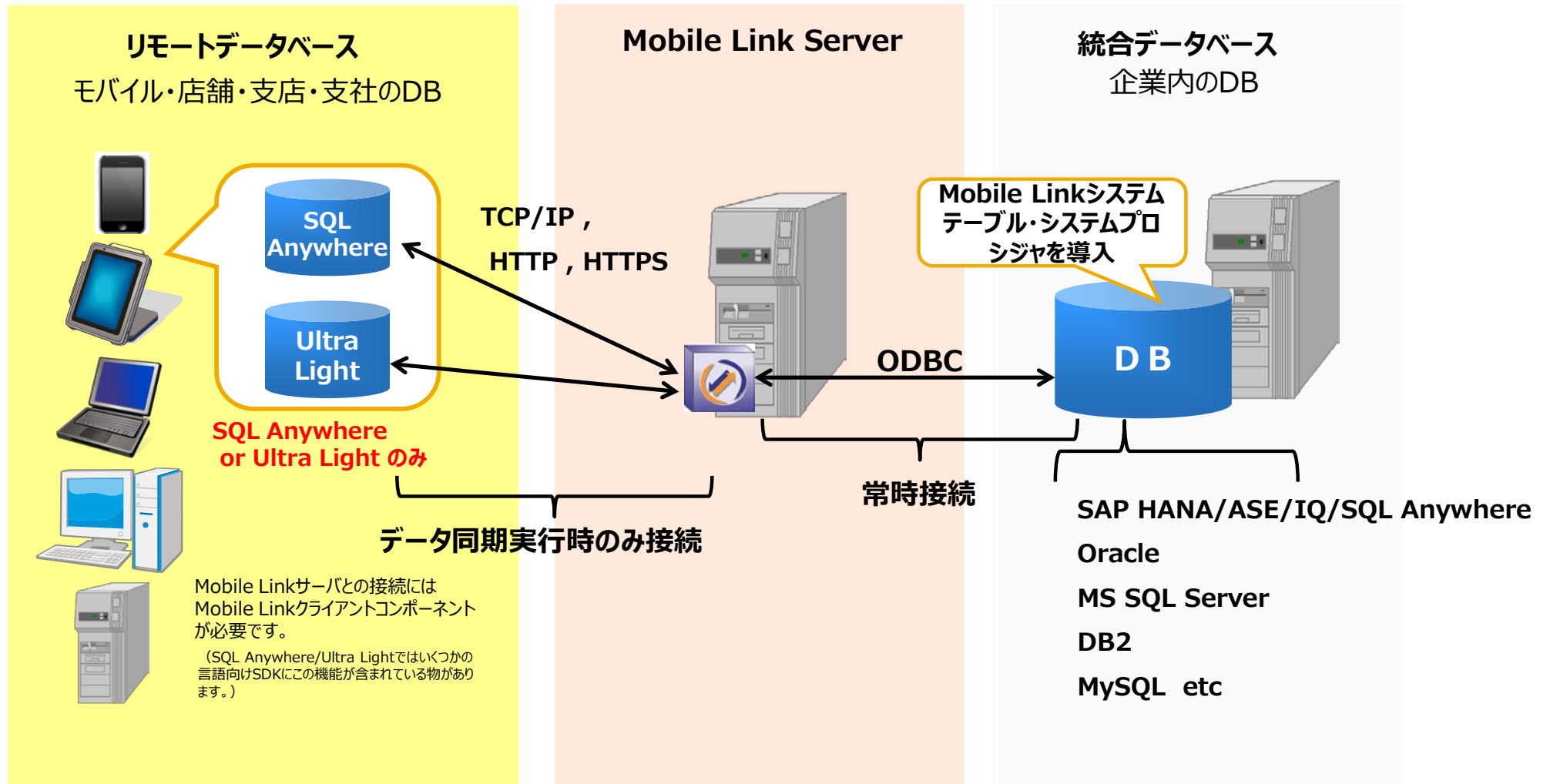
Mobile Linkとは？



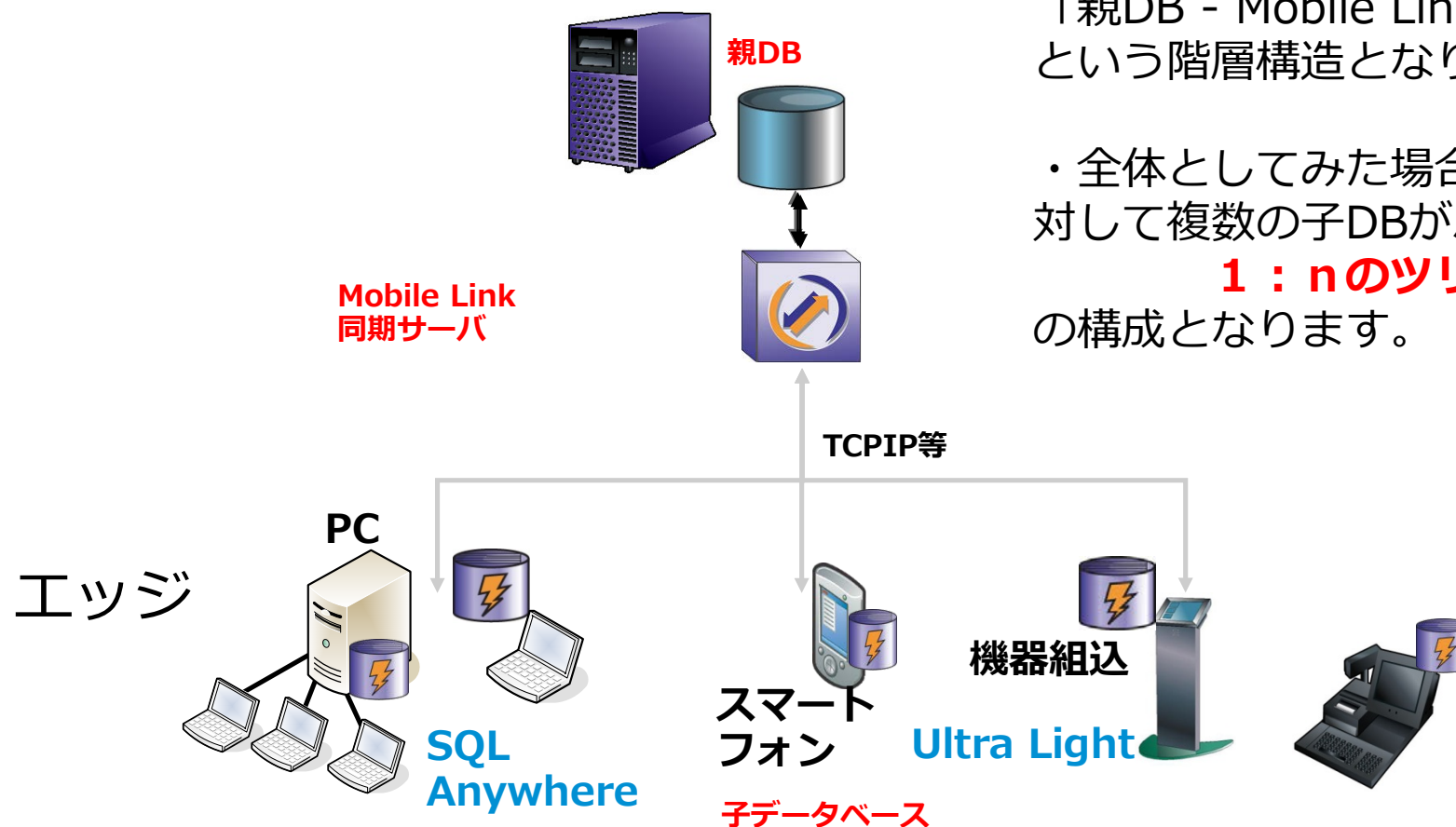
- **DB全体のレプリケーションではない**
- **設定されたテーブルの必要なデータをレプリケーション**
 - レプリケートはテーブル単位で設定。レプリケートするテーブルは名称が違っていても良い。構造が違っていても良い。（レプリケート対象テーブルを設定後、カラムの結びつけを行う）
 - レプリケートは双方向と1方向（デバイス→サーバー、サーバー→デバイス）で3種類から選べる。

Mobile Linkシステムオーバービュー

-RDBMSとの連携



構造 全体オーバービュー



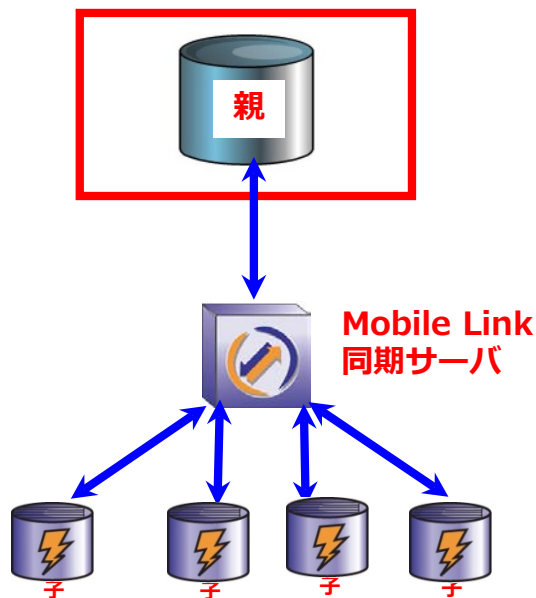
・ Mobile Linkを使用した同期構成は「親DB - Mobile Link同期サーバ - 子DB」という階層構造となります。

・ 全体としてみた場合、左の図に示すように親DB 1 に対して複数の子DBがぶら下がる
1 : nのツリー構造
の構成となります。

統合データベース

統合データベース

- リモートデータベースに対して配信するデータが入った、あるいはリモートデータベースで行ったデータ変更を受け取るデータベースを指します。（親となるデータベース）



- SAP HANA、Oracle、SQL Server、DB2UDB、MySQL、SAP ASE、SAP IQ、SQL Anywhereデータベースを統合DBとして用いることができます。

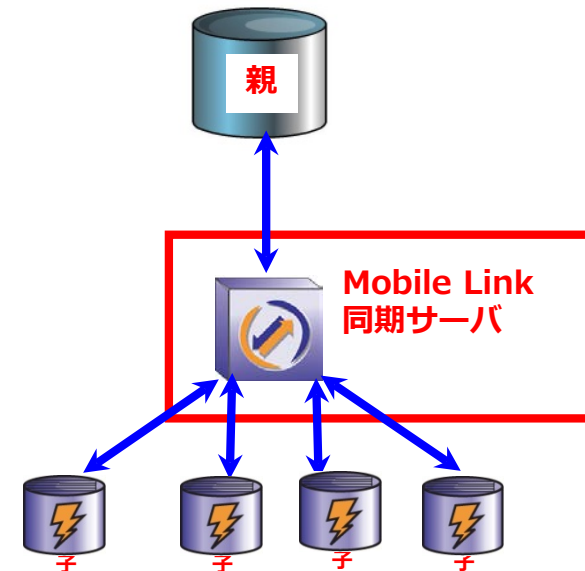
注) 対応プラットフォームやバージョンは弊社HPかマニュアルを参照してください。

- 統合データベースには同期対象となるテーブルの他にMobile Link同期サーバが使用するテーブルやプロシジャ等も配置する必要があります。

Mobile Link 同期サーバ

■ Mobile Link 同期サーバ

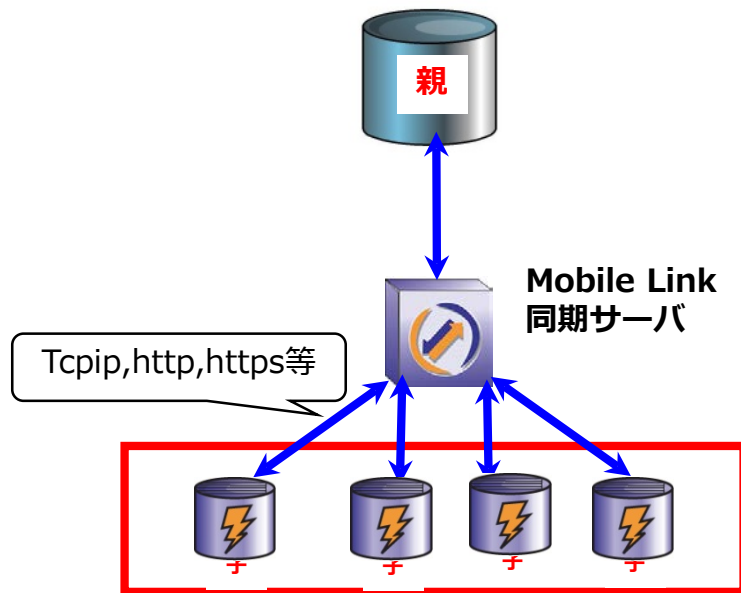
- 子からの同期要求を受け付け、統合DBへ接続し、データの同期処理するサーバプロセス・サービスです。
 - Mobile Linkの場合は統合DBが動作しているサーバマシンで動作させても、別にハードウェアを用意してその上で単独動作させても構いません。
- 動作には統合DBにMLシステム情報を格納する領域が必要です。
 - データ同期の定義や管理情報などを保管



リモート・データベース

リモート・データベース

- 統合データベースからデータを受け取る、あるいは統合データベースに対してデータを送る「子」となるデータベースです。

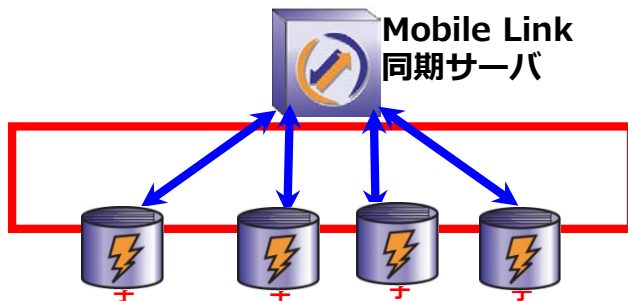


- SQL Anywhere または Ultra Light のみリモート・データベースとして使用することが可能です。
- リモート・データベースは統合DBに“送信”するデータを生成するために、（前回の同期から）同期を行うテーブルに行われたデータの変更を全て記録し、抽出出来る機構が組み込まれています。
 - SQL Anywhereはログ機構
 - Ultra Lightはログカウンタ（行にフラグ）

Mobile Linkクライアント

Mobile Linkクライアント

- Mobile Linkサーバに接続するためのユーティリティ、あるいはAPIです。
 - **クライアントアプリケーション内でクライアントAPIを利用してMobile Linkサーバとの接続、同期の実行を行うことができます。＝クライアント側起動**
 - コマンドプロンプト等から同期を実行するためのユーティリティも一部のプラットフォーム向けに用意されています。
 - SQL Anywhereリモートデータベース用：dbmlsync (.exe)
 - Ultra Lightリモートデータベース用：ulsync (.exe)
 - 同期のテストに非常に便利です。
- リモートDB上でSQLとしてSYNCRONIZE文を実行してリモートDBのSQL Anywhere・Ultralightから同期も可能です。
- リモートDB自体もMobile Linkクライアントの機能を持つが、途中経過等の情報が取得できないという難点がある。

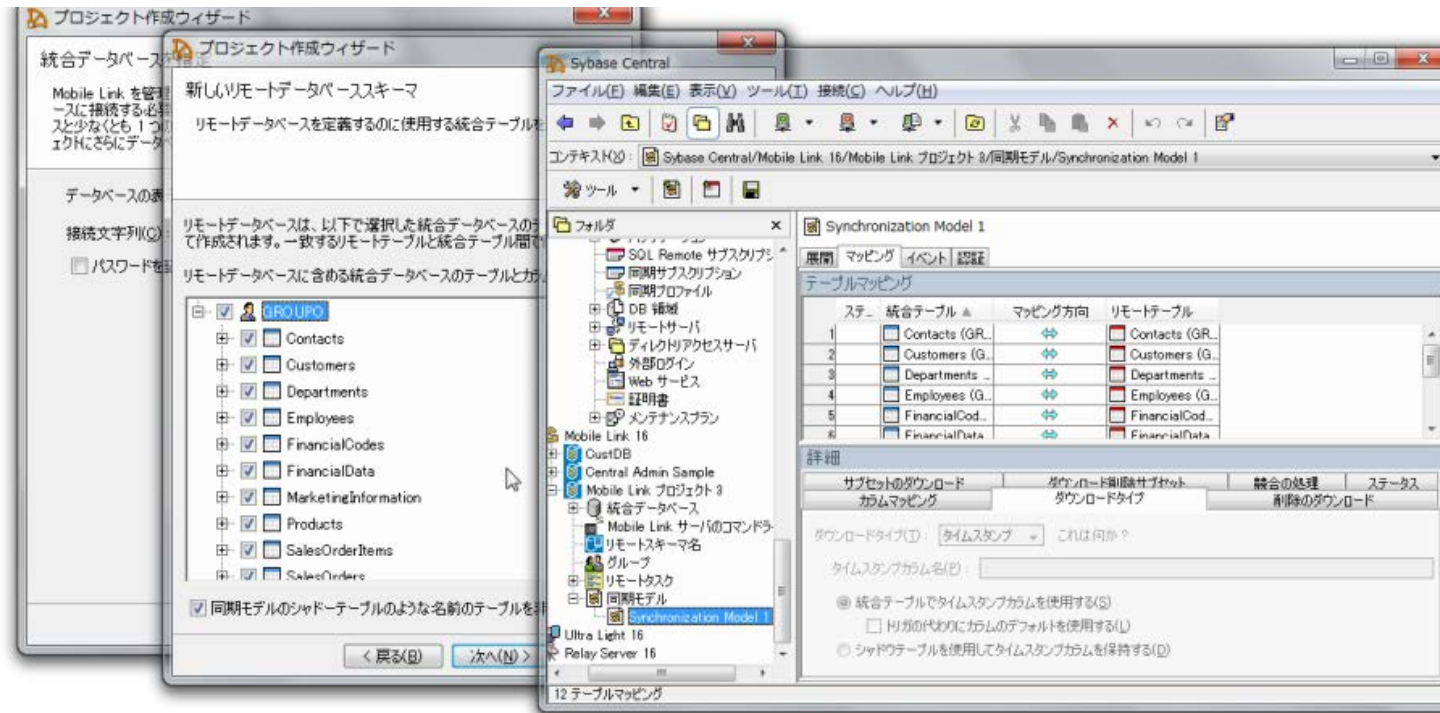


アプリケーション開発用APIに同期サーバと接続する為のクライアント機能が含まれる。

Mobile Link 同期設定ウィザード

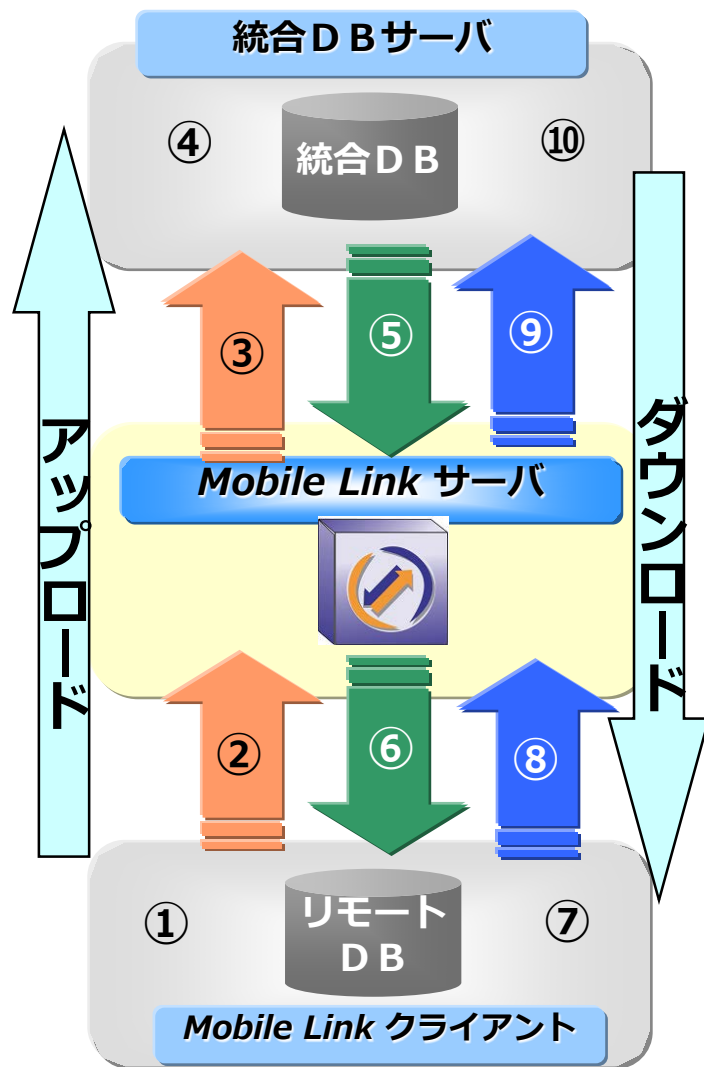
同期設定はウィザード形式のツールにて設定できます。

同期タイプや競合ロジック等の同期設定はウィザード上で選択方式で設定



※内部的にはMobile Linkの同期定義は「スクリプト」です。ウィザードでの設定はスクリプトに変換されて同期定義として保管されます。ウィザードで大まかなロジックを作成し、生成されたスクリプトを手編集ということも可能です。RD Syncは現在このウィザードが提供されていないため、スクリプトを手動記述する必要があります。

データ同期の大まかな手順



パブリケーションに
含まれているテーブ
ル全てに行う

パブリケーションに
含まれているテーブ
ル全てに行う

パブリケーション

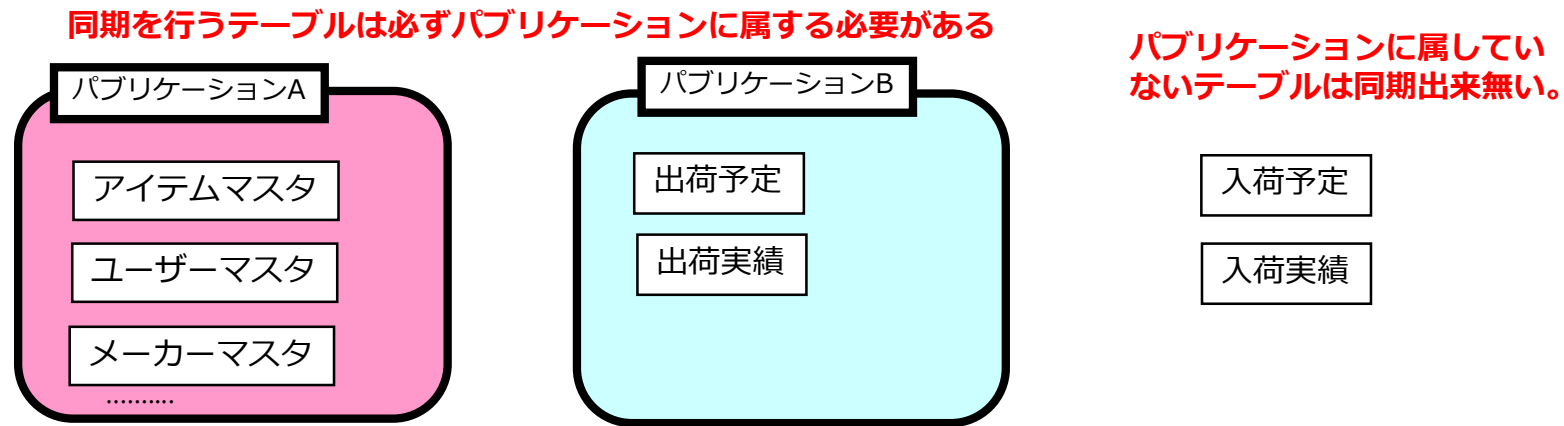
パブリケーション：リモートDB側のテーブル公開設定兼グループ設定

同期対象となる複数のテーブル/カラムに対して割り当てるグループ名。

- 「～～というテーブルを同期したい」という場合、**同期命令にはそのテーブルが含まれるパブリケーションを指定します。**
- 同期時は指定されたパブリケーションに属する全てのテーブルの同期が一度に行われます。

複数定義可。複数のパブリケーションを同期時に同時指定も可能です。

- 同期を行うテーブルは必ず何らかのパブリケーションに属する必要があります。
- ひとつのテーブルはひとつのパブリケーションに属するようにしてください。
 - o 2つのパブリケーションに属する等があるとパフォーマンスが悪くなる場合があります。
- 1テーブル1パブリケーションも可能ですが、大量のパブリケーションはパフォーマンスが悪くなる場合があります。
 - o パブリケーションの処理はリモート側なのでリモートDBのHW性能に依存、要テスト



一般には「一度に同期させたいテーブル類」を1つのパブリケーションに含める。

同期スクリプト

- 同期のルールはテーブルと同期の作業内容、イベント毎にスクリプトで設定
- SQLの {} で囲んである部分をMobile Linkサーバが置換して統合DBで実行
 - Mobile LinkではJava/.NETで記述したAPIを指定することもできる。
- SQLなので統合DBによって同じ同期の定義でもスクリプトに違いが出る。
- スクリプト自体は専用のストアードプロシージャで統合DBに登録する。

master (DBA): download_cursor

```
SELECT "DBA"."master"."sensor_no",  
       "DBA"."master"."description"  
FROM "DBA"."master"  
WHERE "DBA"."master"."last_modified" >= {ml s.last_table_download}
```

master (DBA): download_delete_cursor

```
SELECT "DBA"."master_del"."sensor_no"  
FROM "DBA"."master_del"  
WHERE "DBA"."master_del"."last_modified" >= {ml s.last_table_download}
```

master (DBA): upload_delete

```
/* 統合データベースからローを削除します。 */  
DELETE FROM "DBA"."master"  
WHERE "sensor_no" = {ml r."sensor_no"}
```

master (DBA): upload_insert

```
/* ローを統合データベースに挿入します。 */  
INSERT INTO "DBA"."master" ( "sensor_no", "description" )  
VALUES ( {ml r."sensor_no"}, {ml r."description"} )
```

master (DBA): upload_update

```
/* 統合データベース内のローを更新します。 */  
UPDATE "DBA"."master"  
SET "description" = {ml r."description"}  
WHERE "sensor_no" = {ml r."sensor_no"}
```

Thank you.

Contact information:

F name L name

Title

Address

Phone number

Partner logo

THE BEST RUN 