

# Oracle Direct Seminar



**ORACLE®**

**簡単データ連携!!**

**Oracle Data Integratorの実運用ノウハウ**

日本オラクル株式会社

**Oracle Direct**



# 目次

1. Oracle Data Integrator概要
2. Oracle Data Integrator利用シーン
3. データ連携設計手順
4. お客様事例
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator運用ノウハウ



# 目次

1. Oracle Data Integrator**概要**
2. Oracle Data Integrator**利用シーン**
3. **データ連携設計手順**
4. **お客様事例**
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator**運用ノウハウ**



# Oracle Data Integratorとは

多様なデータソースから異種混合ターゲットへデータ移動と変換を行うETLツール

ETLとは、抽出(Extract) 変換(Transform) ロード(Load)の順でデータ連携する手法

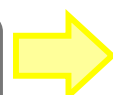
## Oracle Data Integrator (ODI)の特徴

専用サーバー必要なし！



専用の**中間サーバー**が必要なく、パフォーマンスのボトルネックが発生しにくい

簡単お手軽開発



データフローの自動生成、ナレッジ・モジュール等により、**開発工数削減**

柔軟な連携タイミング



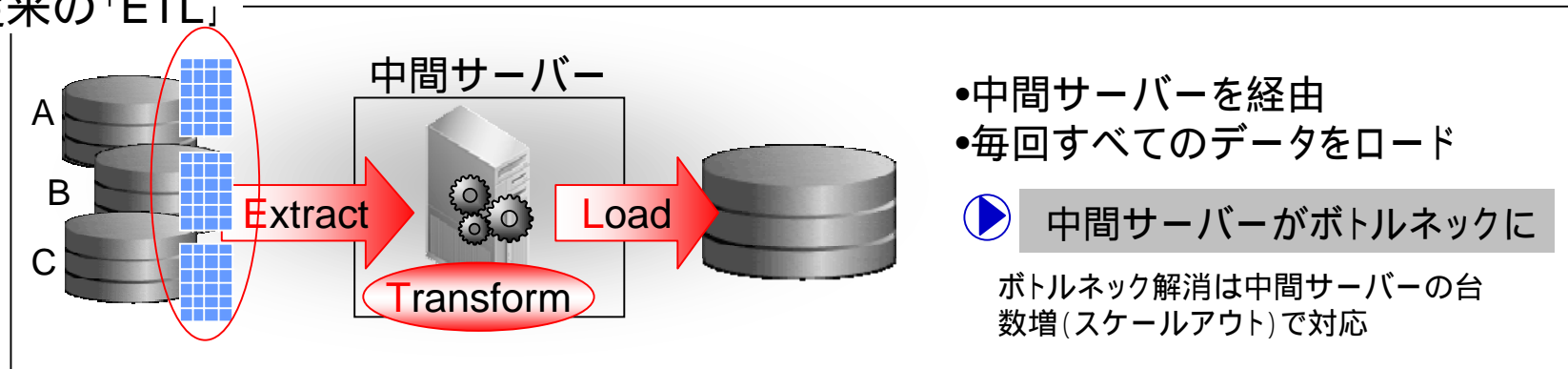
スケジューリングによる連携だけでなく、**リアルタイム連携、SOA連携**に対応

ORACLE

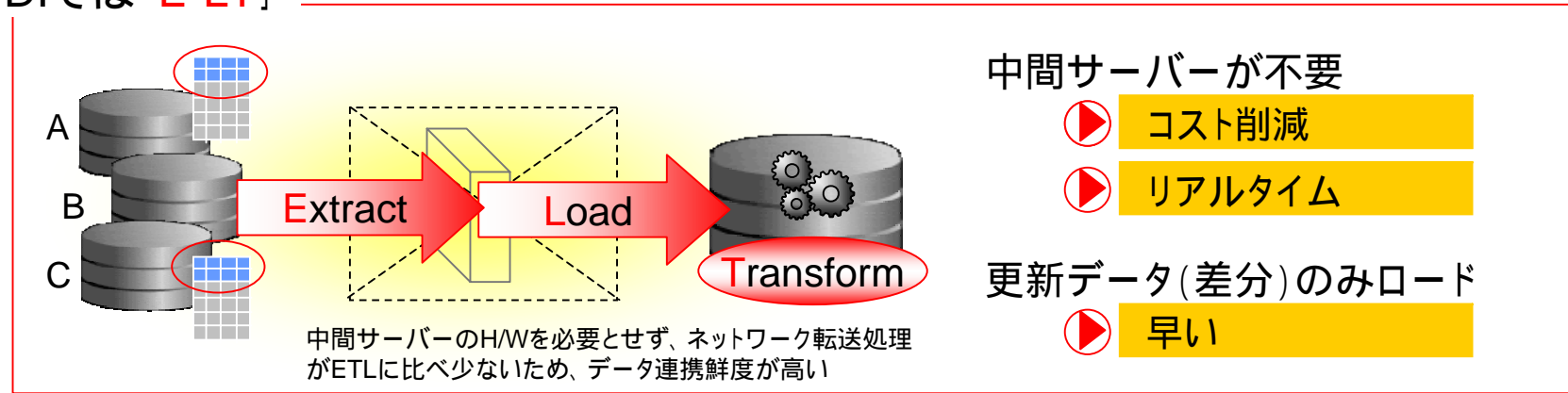
# 中間サーバーを必要としない'E-LT'アーキテクチャ

中間サーバーの排除により、低コスト・高パフォーマンスを実現

従来の「ETL」



ODIでは「E-LT」

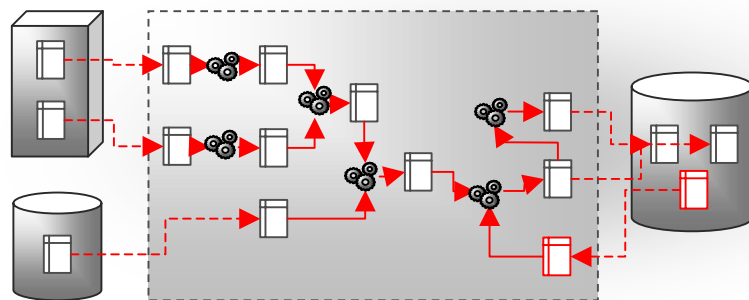


ORACLE

# データフローを自動生成

カスタム・コーディングから、宣言型設計へのシフト

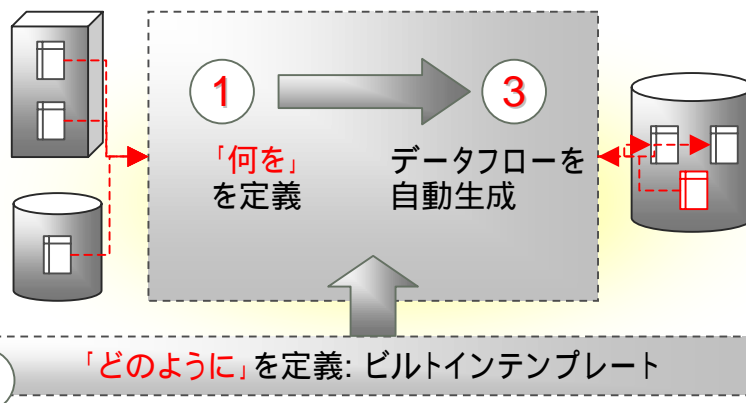
従来の「ETL」設計



- 複雑
- 特殊なスキルが必要

▶ 開発、メンテナンスコストの増加

ODIでは「宣言型設計」



ステップ数の簡素化

- ▶ 開発時間の短縮
- ▶ 習熟期間の短縮

ORACLE

# 直感的なGUI

GUIツールの利用により直感的な開発が可能、高い開発生産性を実現

The screenshot displays the Oracle Data Integrator (ODI) interface. On the left, a tree view shows the project structure. The main workspace contains three source data stores (1 - CUS, 2 - SA, 3 - SP) and one target data store (TRG\_CUSTOMER). A mapping window is open, showing the configuration for the target data store. The mapping window includes a table with columns for the target data store and their corresponding mappings.

開発画面

ソース&ターゲットの一覧

ソースの設定、ソーステーブル内での結合を設定

ターゲットの設定、各カラムのマッピングを設定

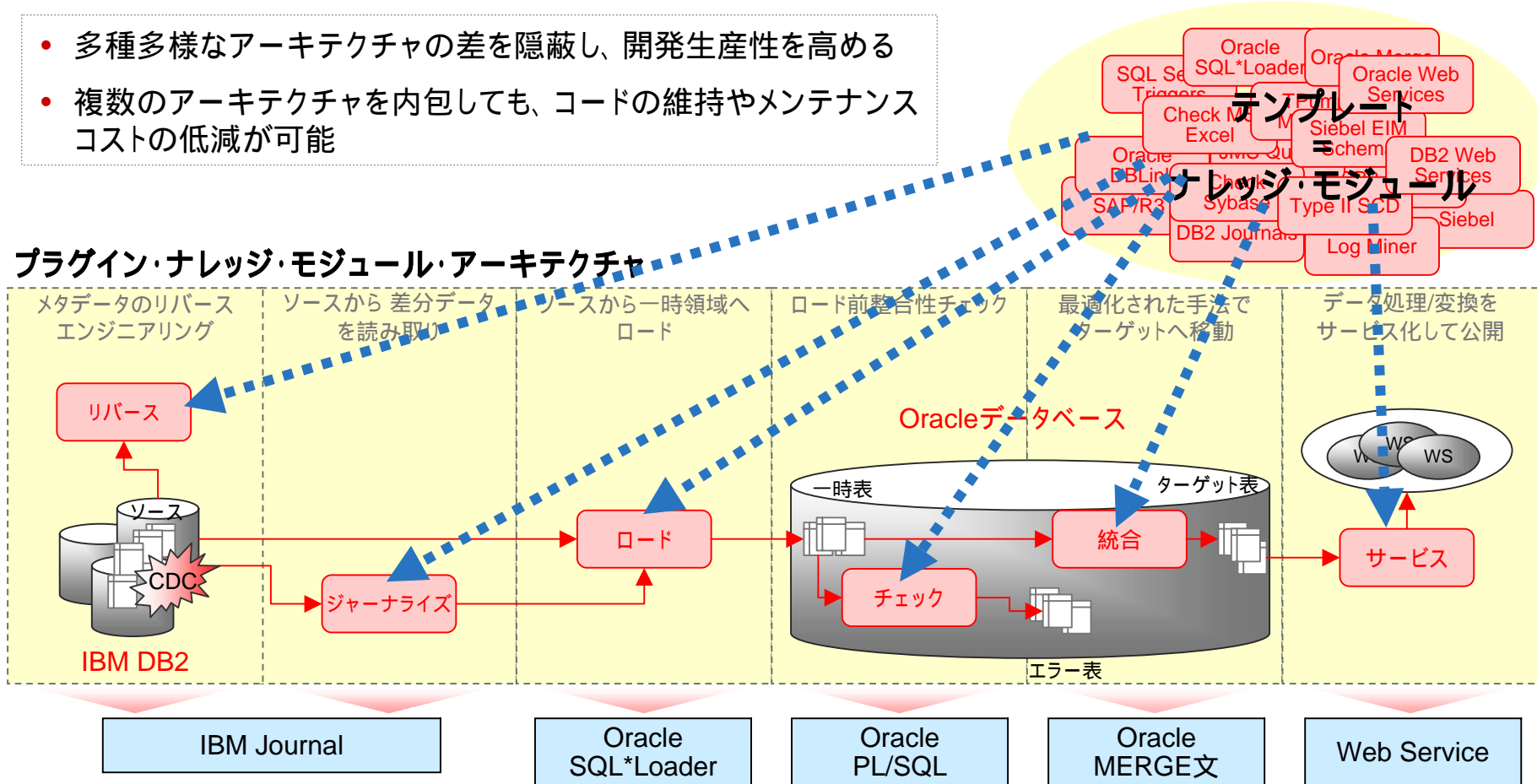
Ind	名前	マッピング
C	CUST_ID	CUS_CUSTID
C	CUST_NAME	For many to many (CUS_FIRST, CUS_LAST)
C	ADDRESS	CUS_ADDRESS
C	CITY	CUS_CITY_ID
C	PHONE	CUS_PHONE
C	AGE_RANGE	SA_AGE_RANGE
C	EMPLOYEE_ID	SP_EMPLOYEE_ID
C	CURRENT_DATE	CUS_CURRENT_DATE

ORACLE

# 事前定義済み設定テンプレートの提供

## テンプレートの再利用による低コスト開発の実現

- 多種多様なアーキテクチャの差を隠蔽し、開発生産性を高める
- 複数のアーキテクチャを内包しても、コードの維持やメンテナンスコストの低減が可能



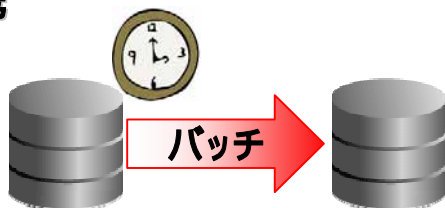
それぞれのアーキテクチャに最適な言語によってコードを生成



# 多様な連携タイミングへの対応

バッチ、リアルタイム連携、イベント駆動化、サービス指向といった幅広い統合要件に対応

## バッチ連携



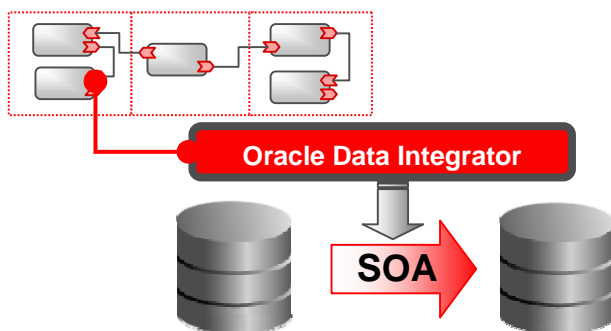
- バッチ処理の定時スケジューリング実行
- Job管理ツールからバッチ処理の起動

## リアルタイム連携



- データ変更をトリガーに、リアルタイムにデータ統合を行うイベント駆動化
- データ変更差分のみを抽出

## SOA連携



- SOAプロセスからのバッチ処理起動
- データ統合フローへの外部SOAサービスの組み込み

# 目次

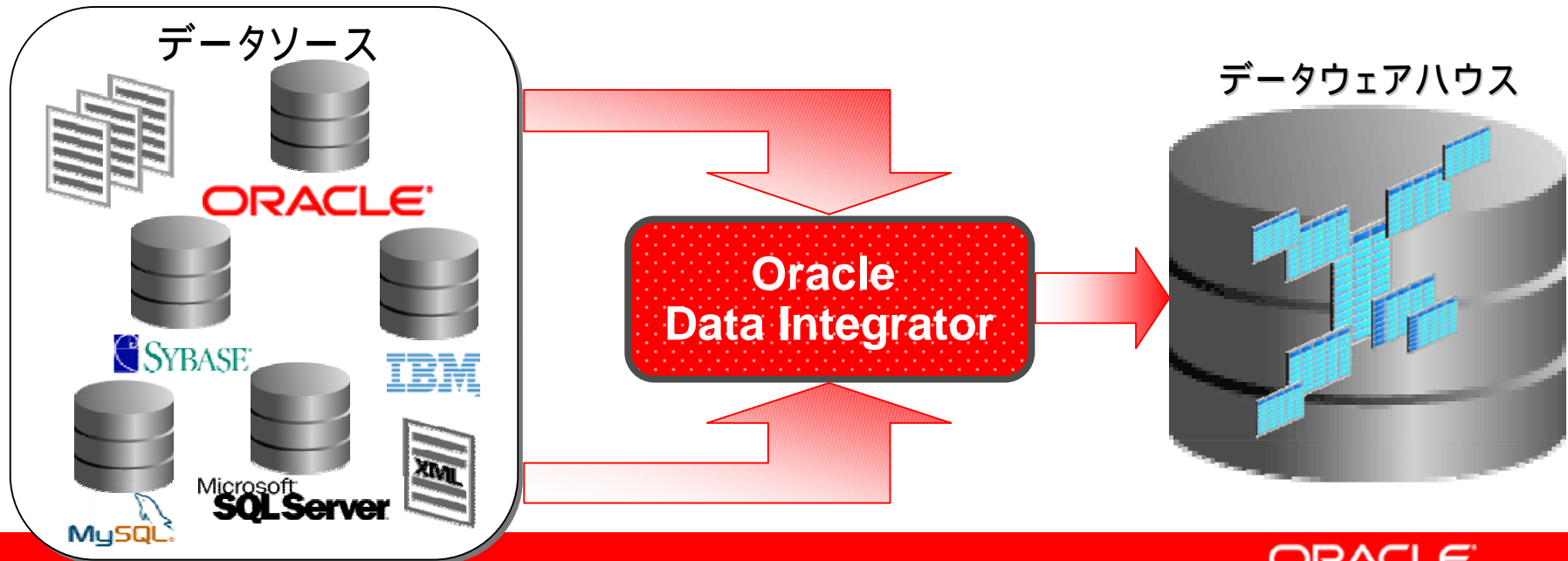
1. Oracle Data Integrator概要
2. Oracle Data Integrator利用シーン
3. データ連携設計手順
4. お客様事例
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator運用ノウハウ



# バッチ連携

- 特長
  - 多種多様なデータソースに対応
  - バルクロード・ユーティリティを使用した**高パフォーマンス**のロード
  - ネットワーク負荷の**軽減**
- 主な適用用途
  - データウェアハウスの構築
  - データ移行(初期移行)
  - スケジューリング(定時バッチ)

(例) 多種多様なデータソースを利用したデータウェアハウスの構築

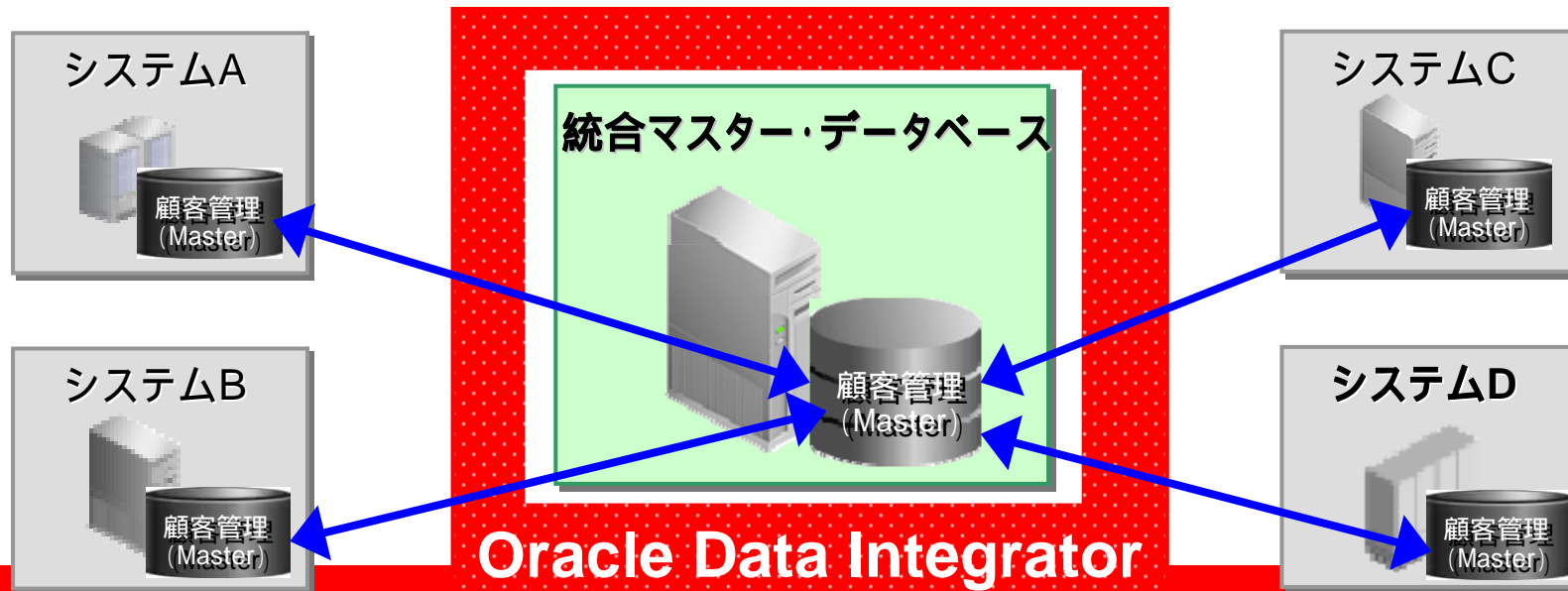


# リアルタイム連携

- 特長
  - 多種多様なデータソースに対応
  - CDC設定時の開発工数の削減
- 主な適用用途
  - マスター・データ管理 (MDM)
  - データ・レプリケーション
  - データ移行時の新旧環境の並行運用
  - リアルタイム・データウェアハウス

CDC:変更されたデータを蓄積する機能

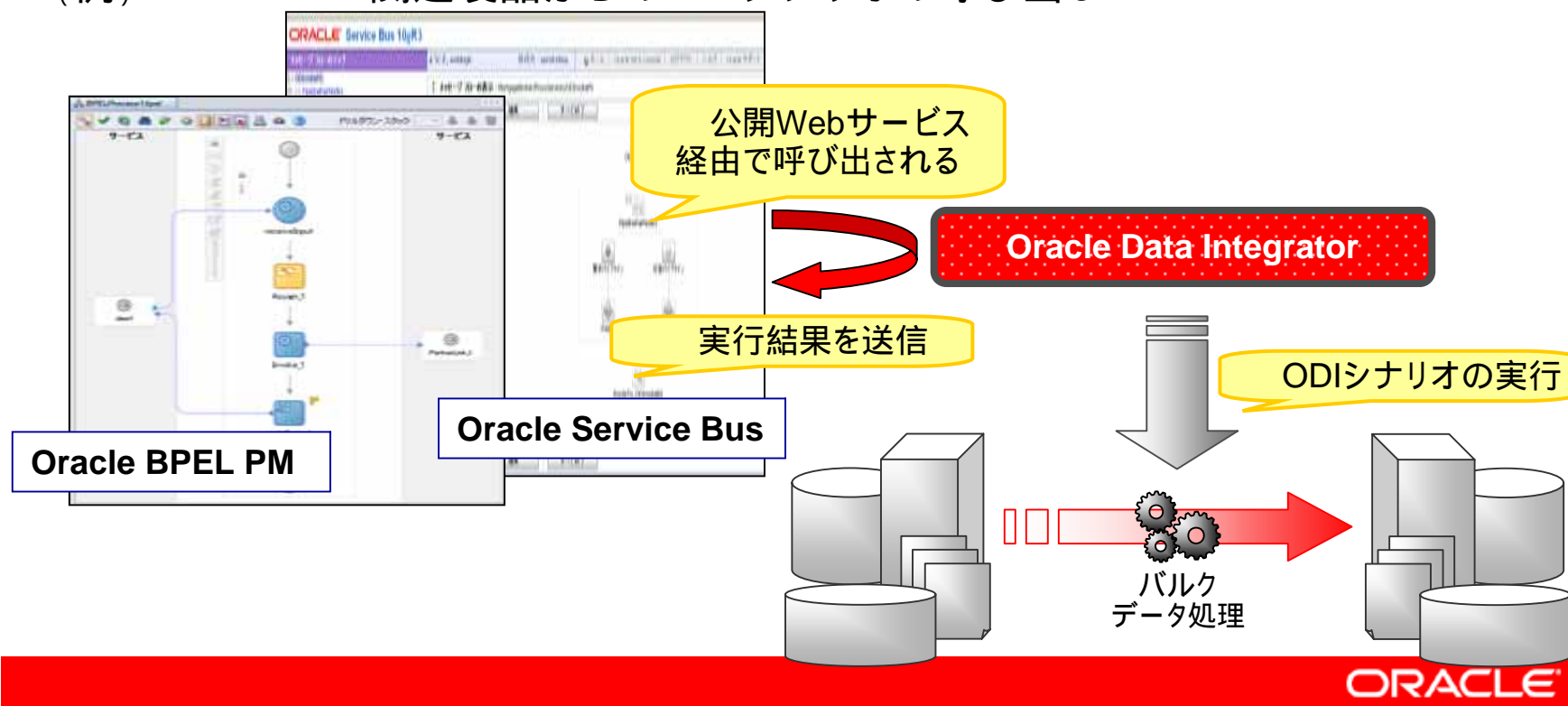
(例) 複数システム間の顧客管理マスターのリアルタイム連携



# SOA連携

- 特長
  - SOAプロセス内の**ボトルネックの解消**
  - Oracle SOA関連製品との**シームレスな連携**
  - 外部SOAサービスをODIのデータ統合フロー内に**組み込み可能**

(例) Oracle SOA関連製品からのODIシナリオの呼び出し

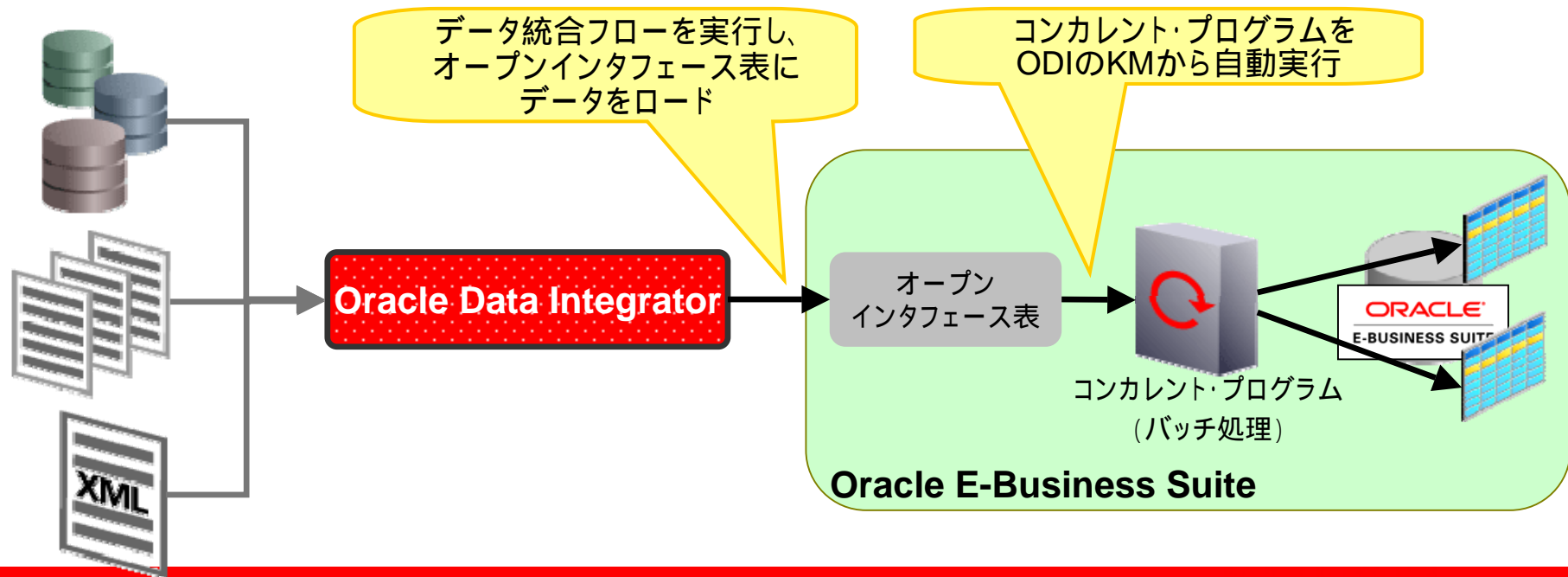


# パッケージ・アプリケーション連携

- 特長

- データ統合フロー作成時の**開発生産性の向上**を実現
- 個々のアプリケーション用のナレッジモジュールを利用してパッケージ・アプリケーションとの**容易な連携**を実現

## (例) Oracle E-Business Suiteとの連携

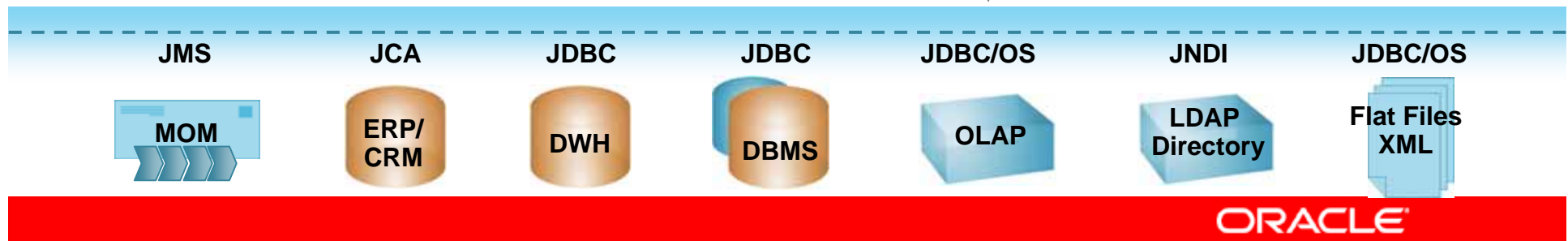
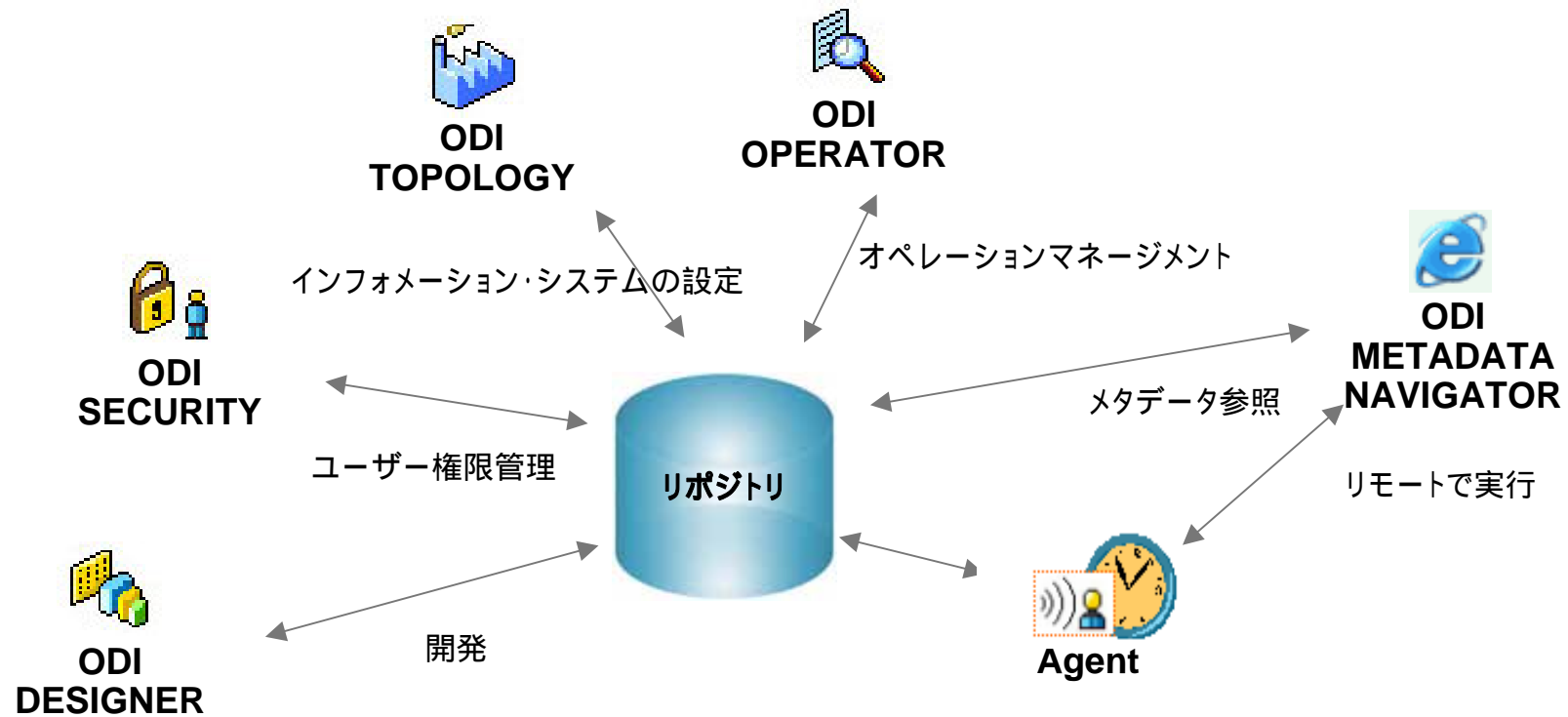


# 目次

1. Oracle Data Integrator概要
2. Oracle Data Integrator利用シーン
3. **データ連携設計手順**
4. お客様事例
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator運用ノウハウ



# ODI コンポーネント



ORACLE



# 「E-LT」アーキテクチャ: ODIの全体構成

## ■ ODIのコンポーネント

ODI Agent (Java process)

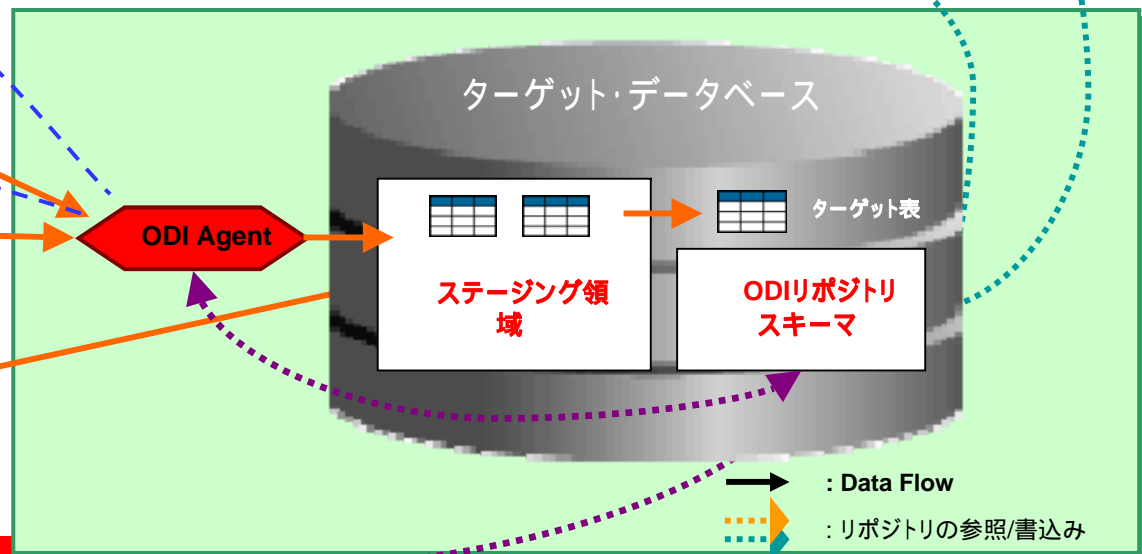
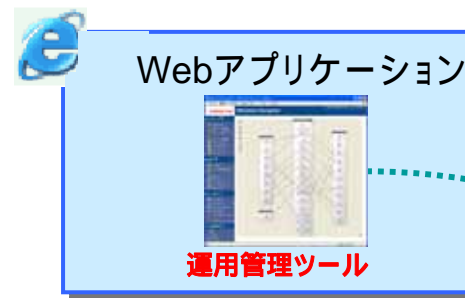
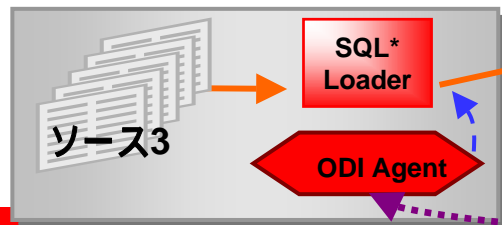
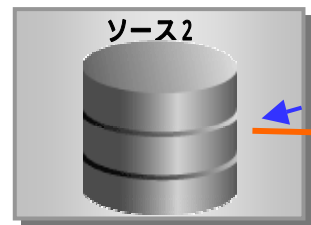
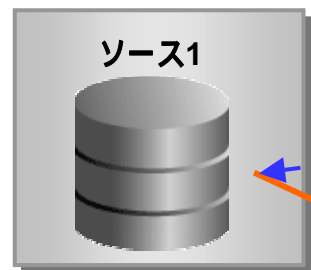
ODI リポジトリ用DBスキーマ

ODI ステージング用DBスキーマ

ODI クライアントツール(開発ツール、運用管理ツール)

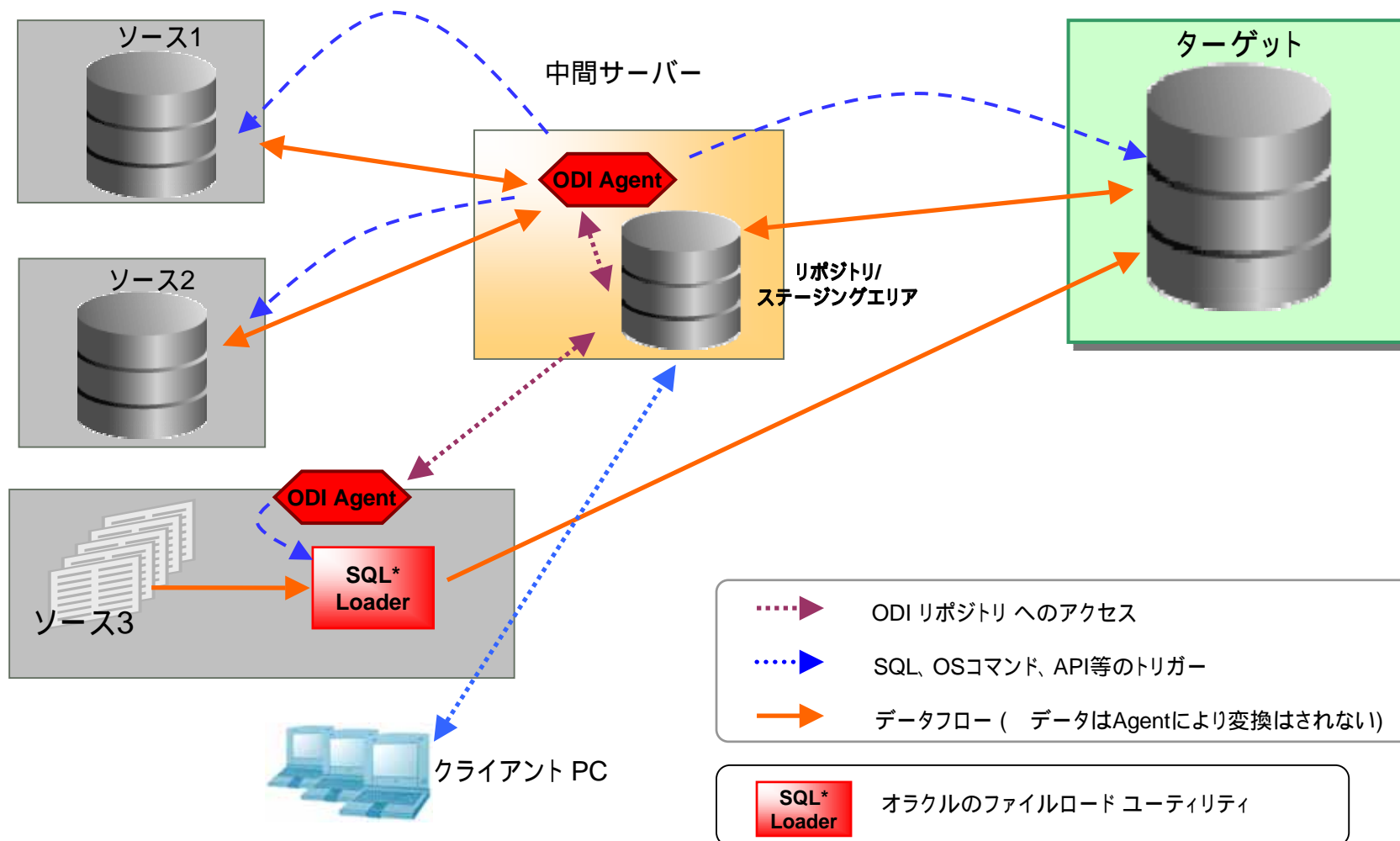
リポジトリやAgentのロケーションは任意

ソース・データベース



→ : Data Flow  
→ : リポジトリの参照/書込み

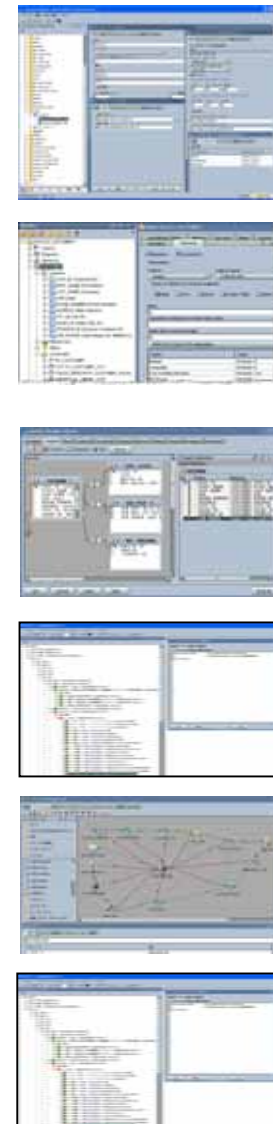
# 中間サーバーを経由してデータ連携をすることも可能



# ODIの開発手順

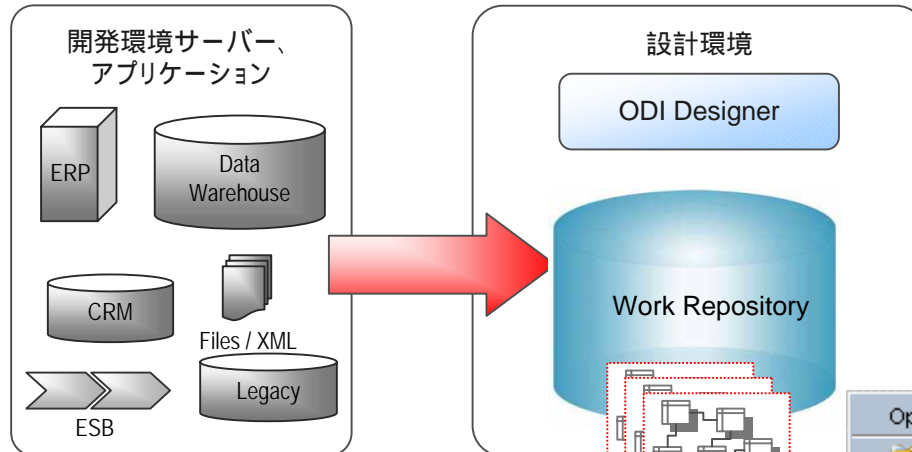
- **アーキテクチャの定義**
  - 物理 & 論理アーキテクチャ
  - コンテキスト
- **メタデータの抽出**
  - テーブル定義
  - 制約
- **インタフェースの設計**
  - ソースとターゲットの選択
  - 変換ルール
  - データ・コントロール(データ品質)
  - ナレッジ・モジュールの選択
- **単体テスト**
  - デバッグ
- **実行フローの設計**
  - パッケージの設計
  - 統合テスト
  - シナリオの生成
  - スケジューリング
- **運用監視**

## 使用するODIモジュール



ORACLE

# メタデータの抽出



## メタデータの抽出

- 自動化
- カスタマイズ可能
- 40以上のテクノロジーをサポート

Condition: New

Definition Control Markers Memo Version Privileges FlexFields

**Datatore: Orders Application - HSQL.SRC\_CITY**

Name

Population Limit

Type

Sunopsis Condition

Where (Use the table alias: SRC\_CITY)

SRC\_CITY.POPULATION > 1000

Message

This city seems to be too small

OK Cancel Apply Help

Open in New Window

Insert Key

Insert Reference

Insert Condition

# インターフェース設計



## 1 「何を」の定義

The screenshot shows the ODI interface with a data flow diagram. A mapping window titled "Mapping: AGE\_RANGE" is open, showing the following SQL code:

```
case
  when AGE_FILE.AGE_RANGE is null then
    UPPER(AGE_FILE.AGE_RANGE)
  else
    'Unknown Age Range for:' || CUSTOMER.AGE
end
```

The mapping window also shows "Execute on:" with options for Source, Staging Area, and Target.

## 3 データフローの自動生成

The screenshot shows the ODI Projects window with a list of Knowledge Modules (KMs) under the "Knowledge Modules" folder:

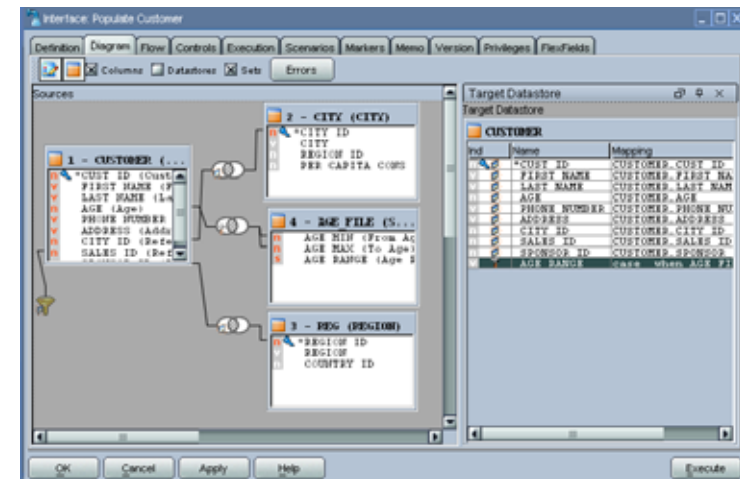
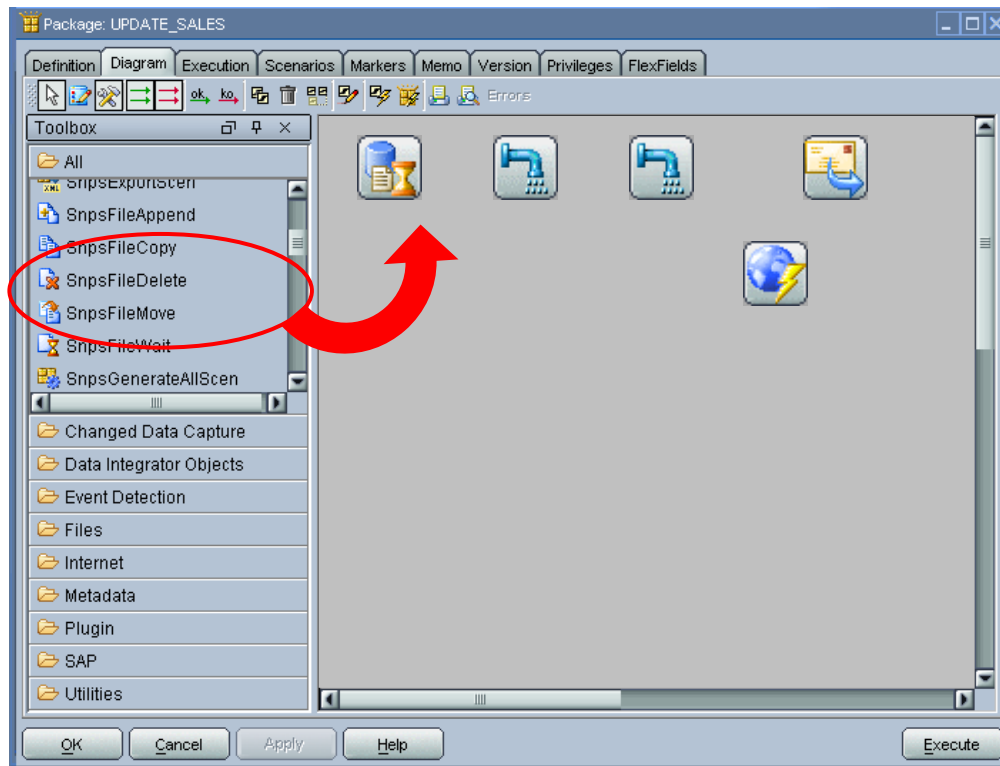
- CKM IBM DB2
- CKM MS SQLServer
- CKM Oracle
- CKM SQL
- CKM Teradata
- IKM File to Teradata (MultiLoad)
- IKM Oracle Incremental Update
- IKM Oracle Incremental Update (MERGE)
- IKM Oracle Incremental Update (PL SQL)
- IKM Oracle Slowly Changing Dimension
- IKM SQL to Teradata (FAST or MULTI LOAD)
- IKM Teradata Incremental Update
- IKM Teradata MultiLoad
- IKM Teradata Replace and Control
- IKM Teradata Slowly Changing Dimension
- IKM Teradata to File (FASTEXPORT)
- JKM DB2 400 Simple
- JKM DB2/400 - Journal Simple
- JKM MSSQL Simple

## 2 「どのように」の定義: テンプレートの選択

# 実行フローの設計

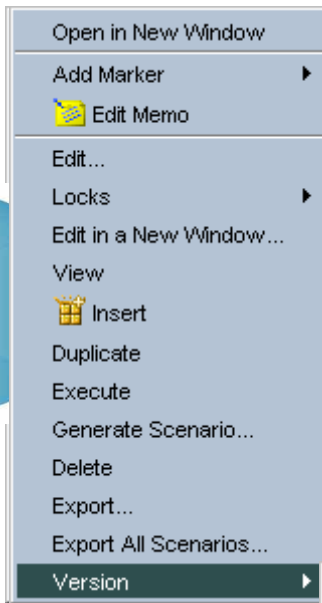
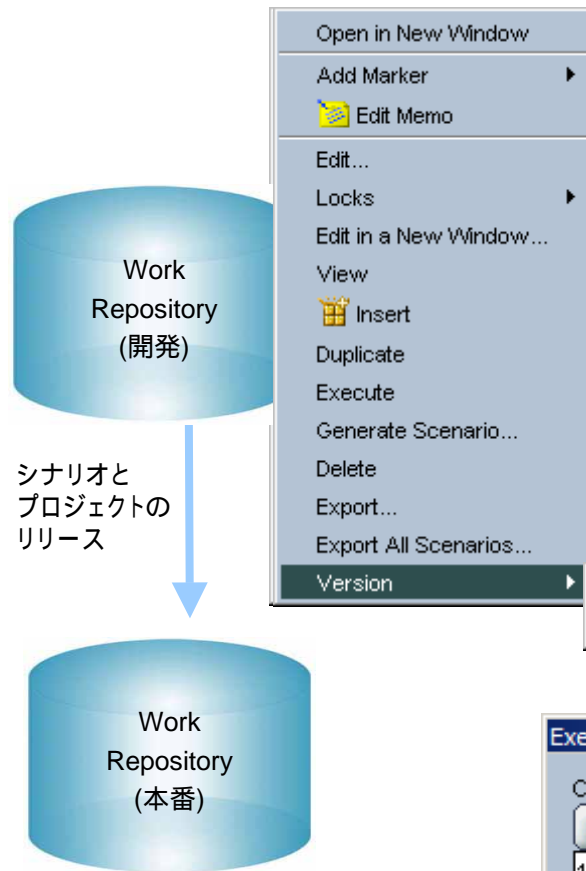


1. 変換の配列
2. OracleDI ツールの活用
  - データ品質プロセス
  - ファイル/アーカイブ管理
  - E-mail 送信/受信
  - Web Service起動
  - イベント検出
  - 独自ツールの作成
3. 制御構造の使用
  - ループ
  - 条件
  - エラーハンドリング



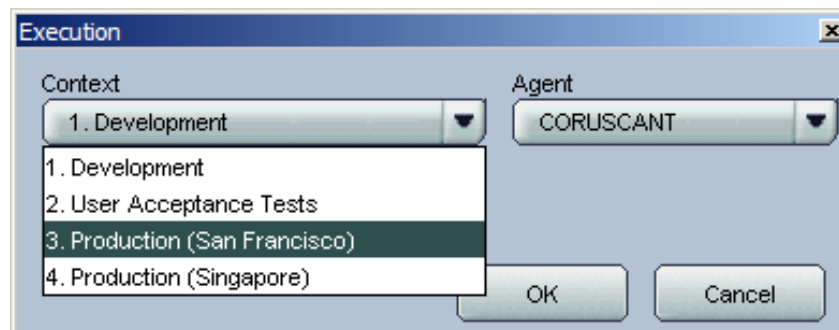
ORACLE

# 実行フローの設計



Current Version: 2.0.1

Version	Date	User	Description
2.0.1	2006-11-23...	SUPERVISOR	Enhancement: Plugged external call to Order Lines Web Service
1.0.1	2006-11-23...	SUPERVISOR	Fix: Added Wait for Changes (CDC)
1.0.0	2006-11-23...	SUPERVISOR	Created



1. シナリオの作成
  - ランタイム用にデータフローのコンパイル
2. データフローのバージョンニング
  - 高度なバージョン管理
3. 本番環境へ配布

# 実行のモニタリング



Start	End	Duration
2006-11-17 15:33:12.54	2006-11-17 15:53:13.82	121

Status	Return Code
Done	0

No. of Rows	No. of Inserts	No. of Updates	No. of Deletes	No. of Errors
127903	102843	25045		15

- 実環境で実行されるセッションの参照
- 生成されたコードのレビュー
- 詳細なランタイム統計
- 失敗セッションの再始動

- Open in New Window
- Edit...
- View
- Stop
- Restart
- Delete



# 目次

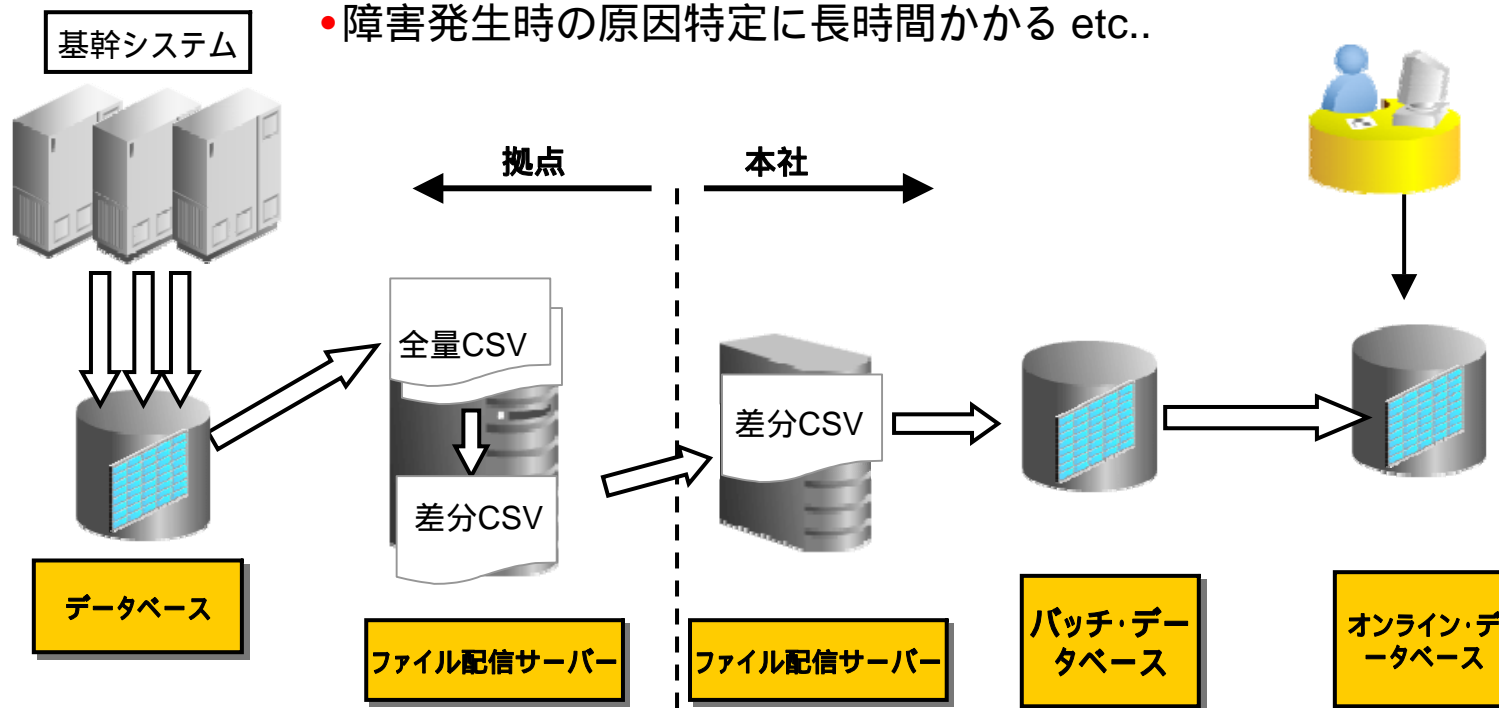
1. Oracle Data Integrator概要
2. Oracle Data Integrator利用シーン
3. データ連携設計手順
4. お客様事例
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator運用ノウハウ



# ODI導入前のシステム構成

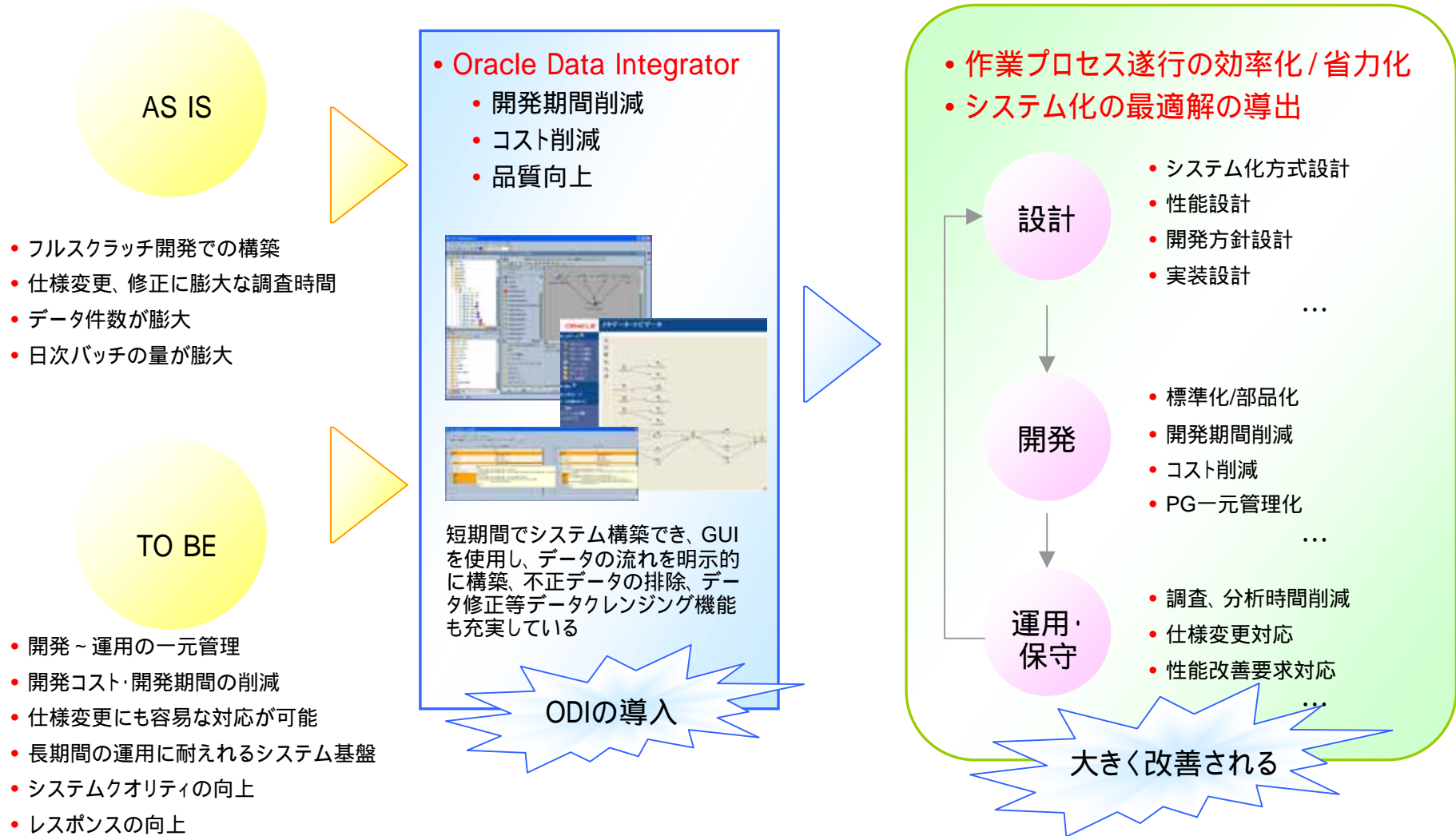
ODI導入前の連携において様々な課題が存在

- フルスクラッチ開発での構築
- 日時バッチの量が増大
- 様々なPGが開発に必要
- 作用変更、修正に膨大な調査時間
- 障害発生時の原因特定に長時間かかる etc..



ORACLE

# 問題解決に向けた方策



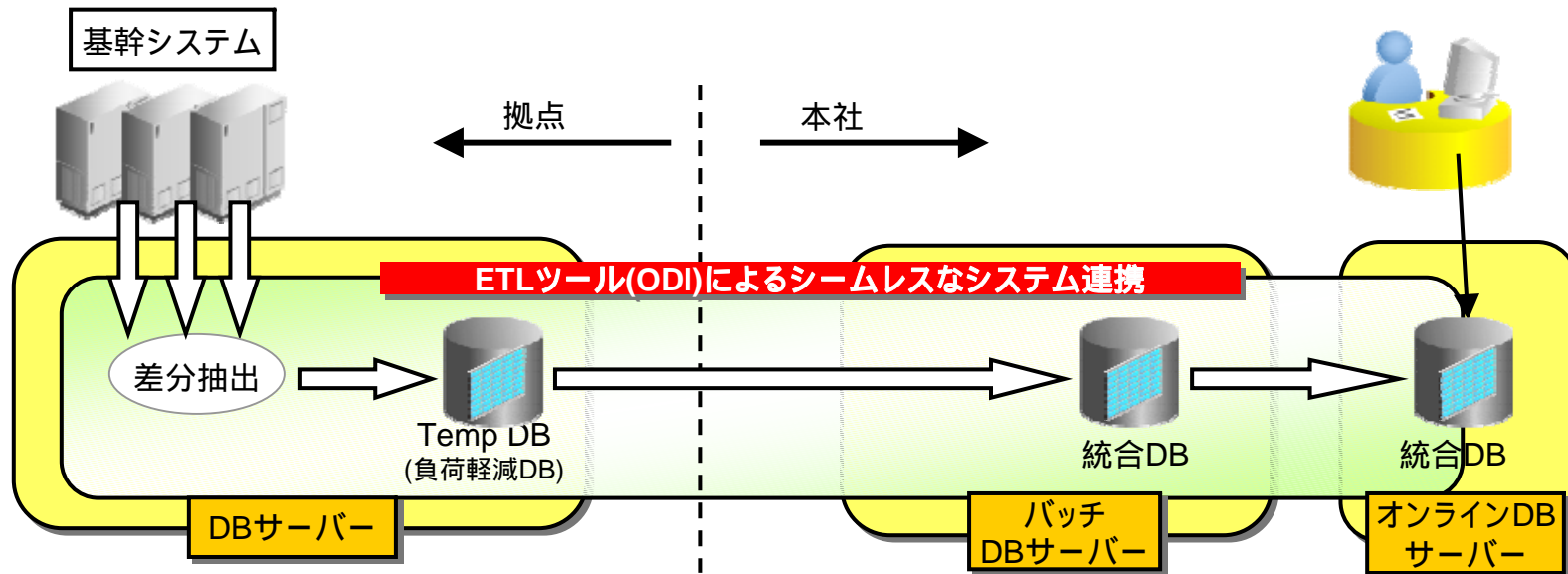
# ODI導入後のシステム構成

ODIの導入で構成がシンプルに!!

旧



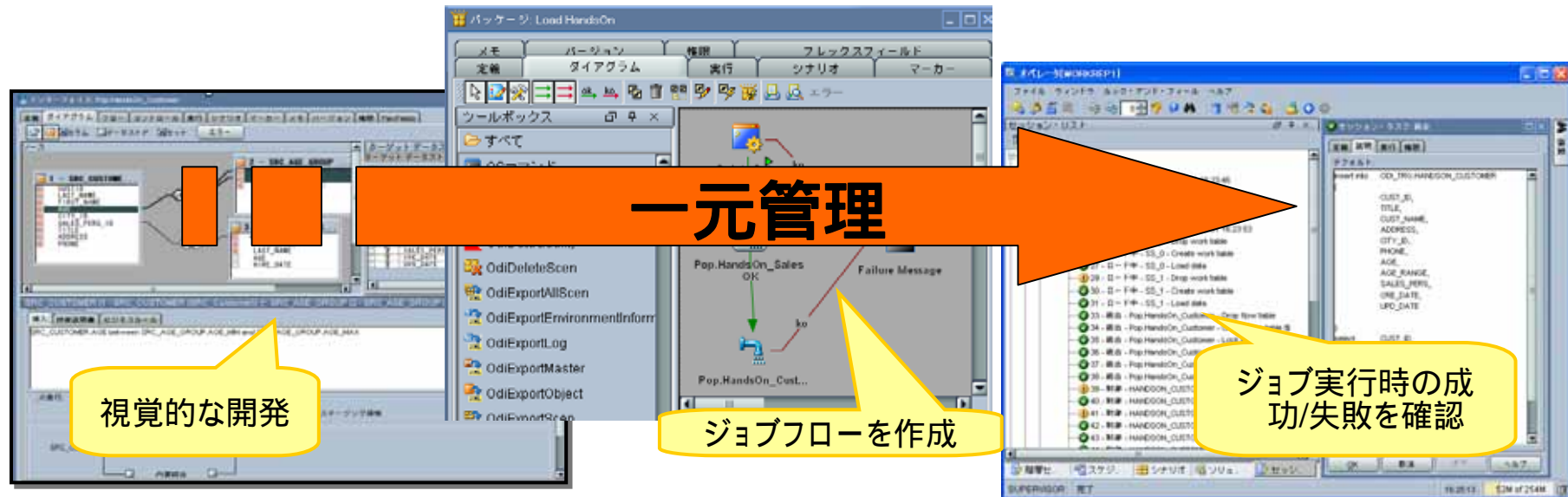
新



ORACLE

# ODIを導入して改善された事

改善点 開発～運用の一元管理化



- ✓ 開発された情報は一元管理されレポジトリ化し管理
- ✓ 開発状況、実行のモニタリングが可能になり、開発から運用にかかる負荷軽減
- ✓ 管理対象ハードウェアを削減

# ODIを導入して改善された事

改善点 開発コスト、開発期間の削減



- ✓ 開発コスト、開発期間の短縮
- ✓ 複数プログラミング言語の習得が不要
- ✓ 開発者の固定化を解消

# ODIを導入して改善された事

改善点 仕様変更にも容易な対応が可能

ドラッグ&ドロップで容易に抽出テーブルを設定できる

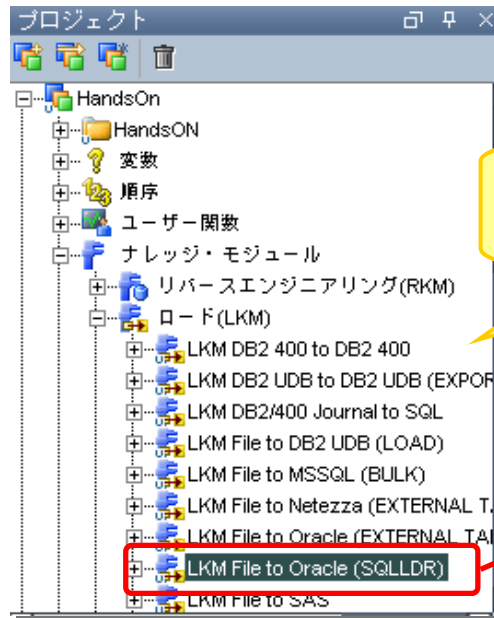
イン...	名前	マッピング
v	*SALES_ID	SRC_SALES.SALES_PERS_ID
v	FIRST_NAME	SRC_SALES.FIRST_NAME
v	LAST_NAME	SRC_SALES.LAST_NAME
v	AGE	SRC_SALES.AGE
v	HIRE_DATE	SRC_SALES.HIRE_DATE

警告  
自動マッピングを実行しますか。  
 この選択を記憶する  
はい いいえ

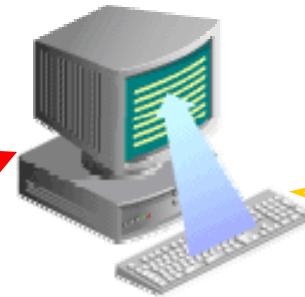
- ✓ GUIベースの開発が行える
- ✓ テーブル仕様の変更に対しても柔軟に対応
- ✓ 設定もパラメータ等で変更可能

# ODIを導入して改善された事

## 改善点 システムクオリティの向上



ナレッジモジュール:  
バッチを使用できる関数群  
(追加、カスタマイズも可能)



ナレッジモジュールを使用  
することで、今まで必要だ  
ったコーディングが不要

- ✓ 定義済みのモジュールを使用可能
- ✓ コード品質が一定以上担保できる
- ✓ ノンプログラミングの為、バグの減少



# 目次

1. Oracle Data Integrator概要
2. Oracle Data Integrator利用シーン
3. データ連携設計手順
4. お客様事例
  - ✓ 複数テクノロジー混在システムのシンプル化に成功
5. Oracle Data Integrator運用ノウハウ





# Oracle Data Integrator 運用ノウハウ

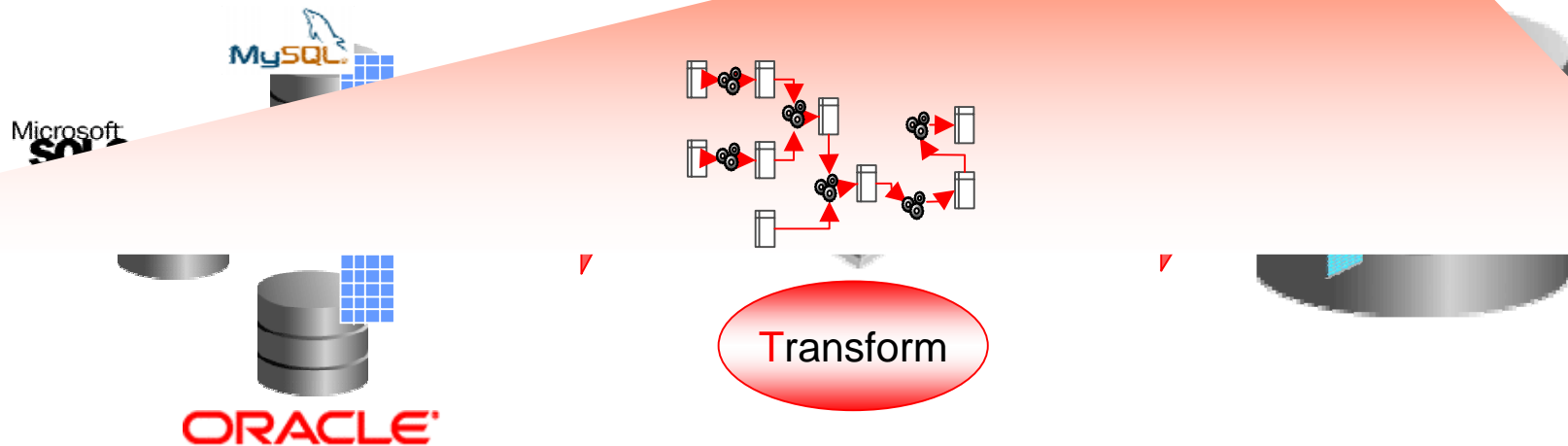
- **構成変更時の対応について**
  - Metadata Navigatorによる可視化
  - バージョン管理
- **障害発生時の対応について**
  - リポジトリ及びAgentの監視
  - Agentの多重化
  - Operatorによる実行管理

# 構成変更時の対応について

従来の連携手法では、

- 構成変更の影響の把握が困難
- 様々なPGを用いて変更作業を行う必要がある
- 変更作業に工数がかかる
- システム管理者間で正確な情報共有が難しい
- 変更時にバグが生じやすい etc..

変更によってどんな影響が出るのかわからない...



ORACLE

# 構成変更時の影響分析

Designer / メタデータ・ナビゲータを用いて構成変更時の影響を正確に把握

The image displays two screenshots from Oracle Metadata Navigator. The left screenshot shows a dependency graph with nodes like REGION, PRODUCT1, ORDER4, ORDER1, and various intermediate tables (IF1, IF2, IF3, IF10, IF20, IF30) leading to IMPRODS. A yellow callout bubble points to this graph with the text: "どの表と表がどんな処理で連携しているのか 一目瞭然!!". The right screenshot shows a model tree for 'CUSTOMER\_ORACLE\_DEMO' with a sub-tree for 'DEMO\_CUSTOMER' highlighted in red. This sub-tree includes nodes for columns, filters, constraints, and data sources like ODI\_DEMO and CUSTOMER\_DEMO.SF. A second yellow callout bubble points to this tree with the text: "表が変更される時、どのプロジェクトにおいてどのように使用されているかドリルダウンし、確認可能".

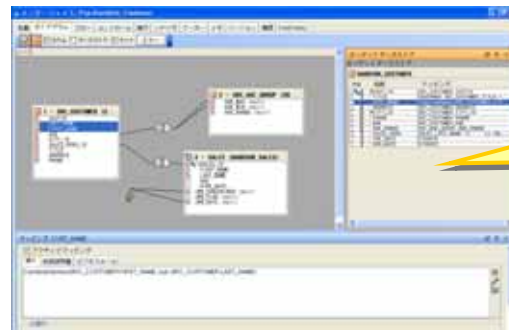
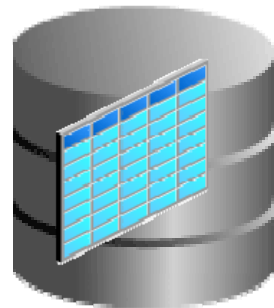
メタデータが一元管理されており、仕様変更の影響範囲が素早く把握できる

# 連携対象の表定義を変更したとき

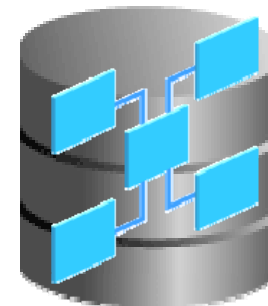
GUI上でメタデータを修正することで、連携方法を再構築せず更新を反映

追加項目が転送対象とならない場合、  
修正は不要

追加項目が転送対象とならない場合、ODIのメタデータ(表定義)に項目(カラム)を追加しマッピングを修正するだけで、既存のデータフローをそのまま利用可



変更作業時間短縮  
変更に伴うバグも発生しにくい



# 連携対象の表定義が変更したとき

ODIのメタデータ(表定義)にカラムを追加し既存データフローのマッピングを修正するだけ

The image illustrates the process of updating ODI metadata when a table definition changes. It shows the 'SOURCE' and 'TARGET' views, the 'ORDER\_DETAIL' mapping table, and the 'ORDERDATA' table definition in two states: before and after a column is added.

**カラム追加前のデータフロー (Before Column Addition):**

イン...	名前	マッピング
Y	PRODUCT_NAME	PRODUCT.PRODUCTNAME
Y	UNIT_PRICE	PRODUCT.UNITPRICE
Y	ORDER_ID	ORDERDATA.ORDER_ID
Y	PRODUCT_NO	ORDERDATA.PRODUCT_NO
Y	AMOUNT	ORDERDATA.AMOUNT
N	TOTAL_PRICE ...	

**ODIのメタデータ(表定義)にカラムを追加 (Add Column to ODI Metadata):**

順序	名前	タイプ	論理長	スケール	NOT NULL
1	ORDER_ID	NUMBER	22	0	<input checked="" type="checkbox"/>
2	CUSTOMER_ID	NUMBER	22	0	<input type="checkbox"/>
3	COMPANY_NAME	VARCHAR2	100	0	<input type="checkbox"/>
4	ADDRESS	VARCHAR2	200	0	<input type="checkbox"/>
5	TEL	VARCHAR2	50	0	<input type="checkbox"/>
6	PRODUCT_NO	VARCHAR2	40	0	<input type="checkbox"/>
7	AMOUNT	NUMBER	22	0	<input type="checkbox"/>
8	UNIT_PRICE	NUMBER	22	0	<input type="checkbox"/>
9	TOTAL_PRICE	NUMBER	22	0	<input type="checkbox"/>

**で追加したカラムにマッピングを定義 (Define Mapping for Added Column):**

イン...	名前	マッピング
Y	ORDER_ID	ORDERDATA.ORDER_ID
Y	PRODUCT_NO	ORDERDATA.PRODUCT_NO
Y	PRODUCT_NAME	PRODUCT.PRODUCTNAME
Y	AMOUNT	ORDERDATA.AMOUNT
Y	UNIT_PRICE	PRODUCT.UNITPRICE
N	TOTAL_PRICE ...	

# ドキュメント作成機能

最新の連携構成情報をPDFファイルにより、情報共有が可能

- 最新の構成情報を共有することで、管理者・開発者間の意思統一が可能
- レポート生成コストを削減
- 出力できる単位は
  - モデル
  - ナレッジ・モジュール
  - パッケージ
  - インタフェース
  - プロシージャ
  - フォルダ

メンバー全員に最新の構成情報を簡単に共有



# マーカー/メモ機能の利用

設計・開発の進行状況をマーカーにより、把握することが可能

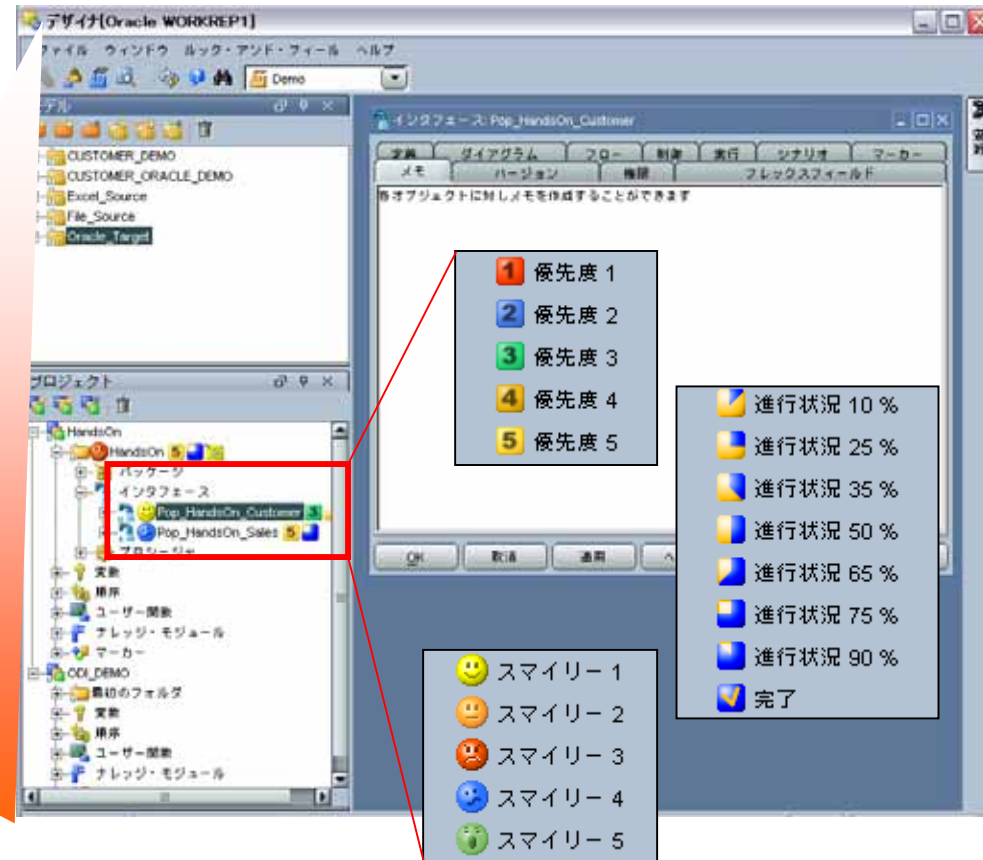
- マーカーを記すことで、開発中にメンバー間で進行状況の共有を図れる

マーカー種類

- 優先度
- 進行状況
- スマイリー

- オブジェクトに対しメモ書きも可能

複数のオブジェクトの進捗状況を一覧で確認できる





# バージョンニング機能

オブジェクトのバックアップをバージョンとして管理し、リストアすることが可能

- 以下のオブジェクトのバージョンニング可能

- プロジェクト
- フォルダ
- パッケージ
- インタフェース
- プロシージャ
- KM
- シーケンス
- シナリオ
- ソリューション
- 変数
- ユーザー関数
- モデル
- モデル・フォルダ
- シナリオ

- 同一オブジェクトのバージョン比較変更を比較できる

- 変更前の状態にロールバック可能

- 変更履歴のPDFファイル出力



やっぱり変更前の状態に戻そう!



色	意味
白	変更なし
赤	削除
緑	追加/新規
黄	変更

ORACLE



# Oracle Data Integrator 運用ノウハウ

- **構成変更時の対応について**
  - Metadata Navigatorによる可視化
  - バージョン管理
- **障害発生時の対応について**
  - リポジトリ及びAgentの監視
  - Agentの多重化
  - Operatorによる実行管理

# ODIによる障害発生時の対応について

マクロ・マイクロ視点で連携の実行を監視し、エラーや障害に対する迅速な検知・特定

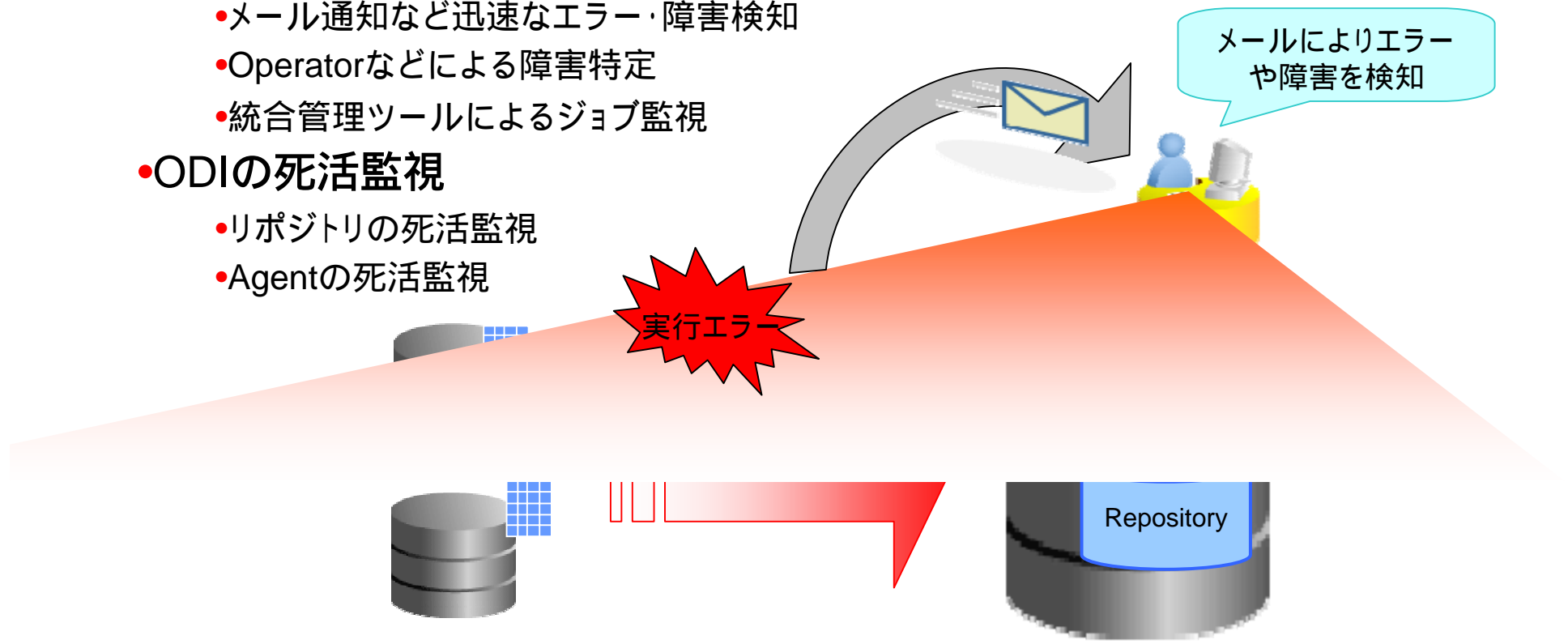
## 可用性を高めるポイント

### •連携全体の実行監視

- メール通知など迅速なエラー・障害検知
- Operatorなどによる障害特定
- 統合管理ツールによるジョブ監視

### •ODIの死活監視

- リポジトリの死活監視
- Agentの死活監視



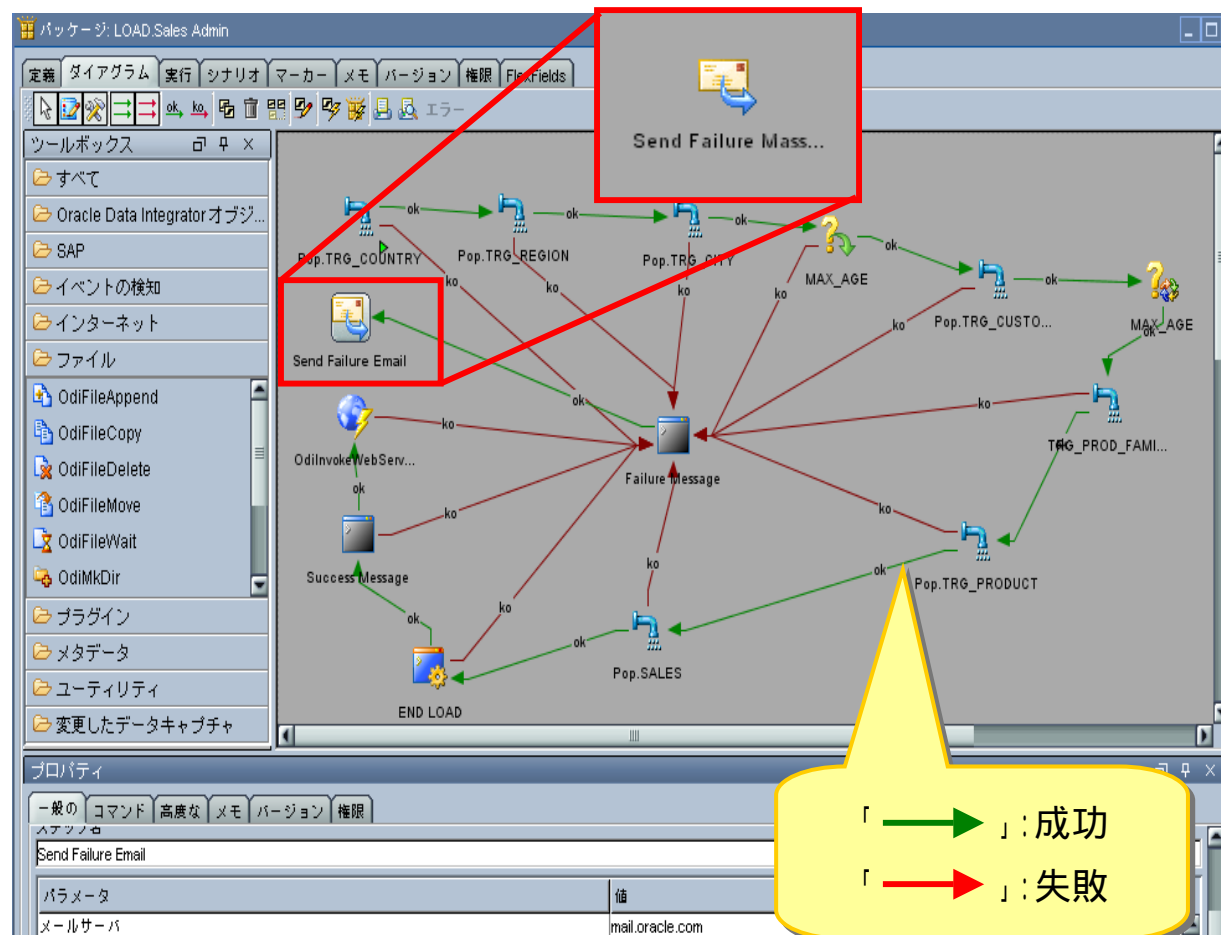
ORACLE

# ジョブフロー中のエラー監視

メール通知によりジョブフロー中にエラーが生じたことを管理者に通知

- パッケージ中にメール通知を行うODI Toolsを埋め込むことでエラーが生じた際に、素早くメール通知

- メール通知だけでなく、エラーが発生した際にOSコマンドを実行させるなどの設定も可能



# ODI のジョブ監視

ODI Operatorにより連携処理の実行結果を参照可能

- GUIベースで停止した場所やエラーの内容を把握し、再実行することが可能
- エラーが生じた実行プロセスをドリルダウンすることで、実行コマンドやエラー情報を確認可能
- Operator上で修正後、再実行可
- メタデータ・ナビゲータからも実行ログを参照可

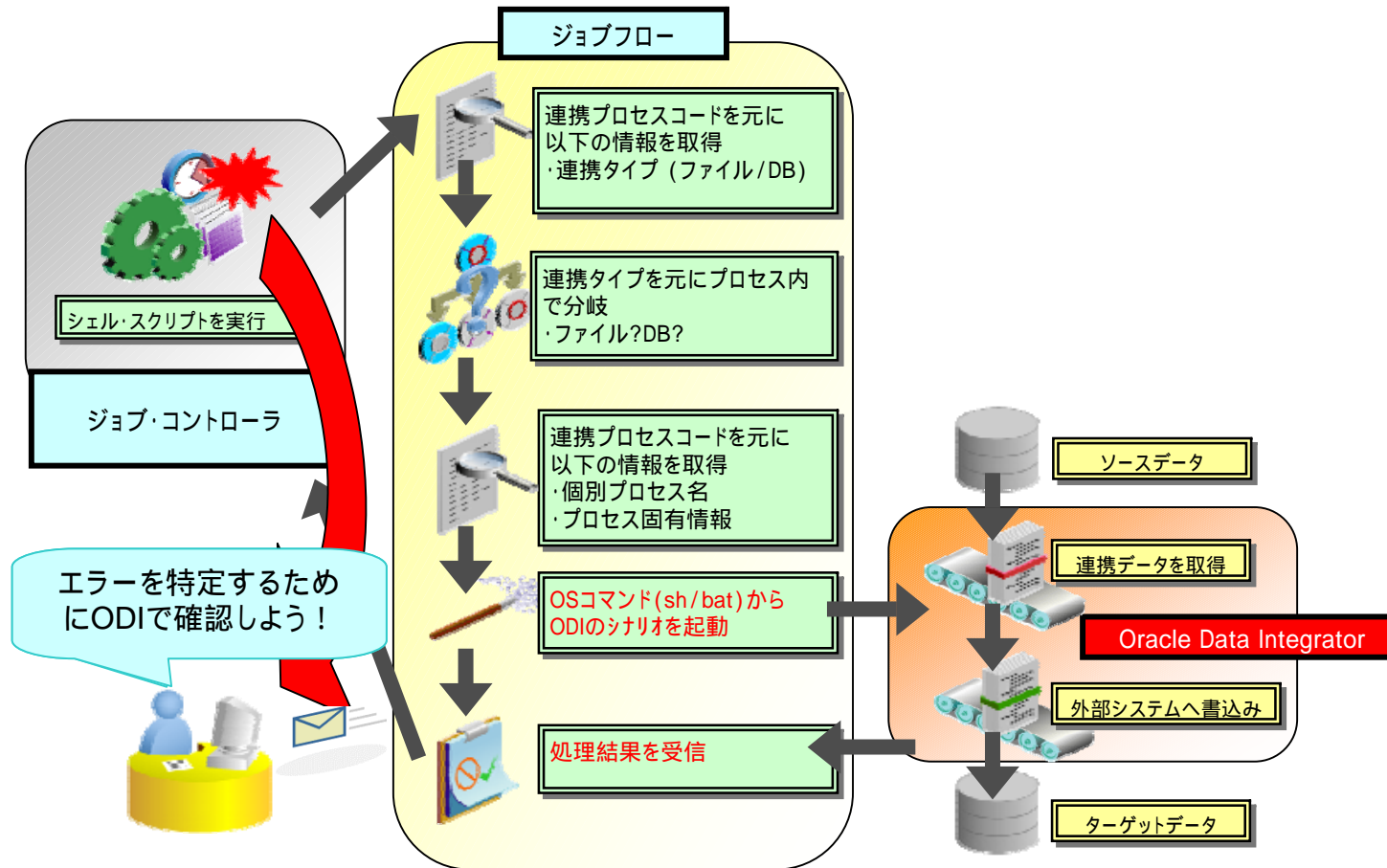
The screenshot displays the ODI Operator interface. On the left, a tree view shows a job execution log with various steps and their statuses. A red box highlights a specific step: '18 - 結合 - JDL\_SHOHLIN - insert check sum into check table'. On the right, a detailed view of this step is shown, including a table with columns for '開始' (Start), '終了' (End), and '種別' (Type). The 'エラー' (Error) dropdown is set to 'B55', and the message area shows the error details: 'B55: 42000: java.sql.SQLException: ORA-00955: すでに使用されているオブジェクト名です。' (ORA-00955: object name already exists).

発行コマンドやエラーの詳細が確認可能

ステータスの確認 (成功 = 緑、実行中 = 黄緑、失敗 = 赤)

# ODI のジョブ監視

統合管理ツールと組み合わせたジョブ監視の実行

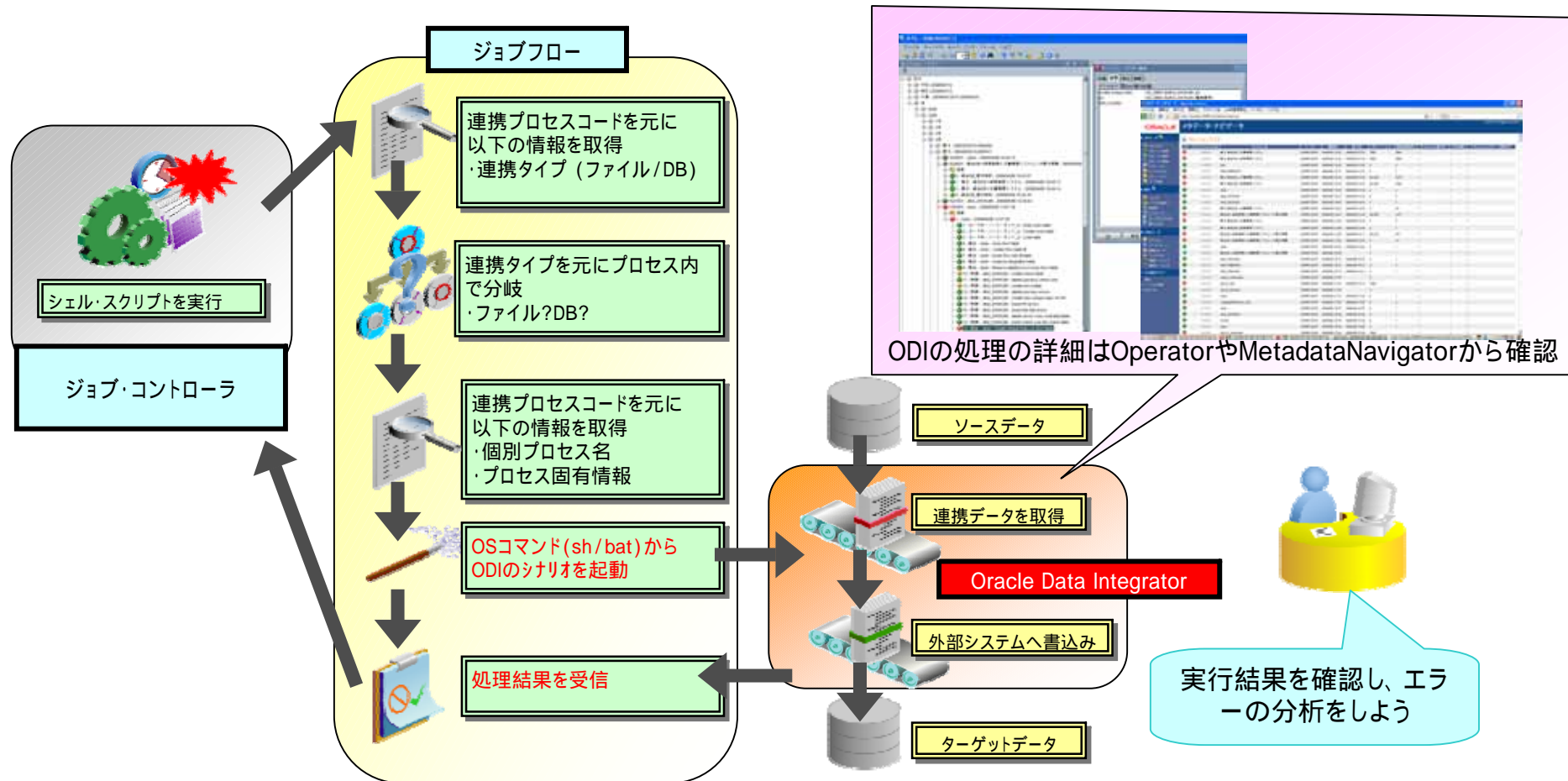


普段は使い慣れた管理ツールを使用し、問題が生じた際にOperatorなどで分析を行える

ORACLE

# ODI のジョブ監視

統合管理ツールと組み合わせたジョブ監視の実行



普段は使い慣れた管理ツールを使用し、問題が生じた際にOperatorなどで分析を行える

ORACLE

# ODI のジョブ監視

参考: ODIのシナリオをOSコマンド(sh/bat)から実行する方法

The image illustrates the process of creating and executing an ODI scenario from the command line. It features several overlapping windows:

- Top Left:** A scenario diagram window titled "insert\_MPN005" showing a flowchart with steps like "insert\_bmp\_dg\_data", "IF1", "IF2", and "IF3".
- Top Right:** A "プロジェクト" (Project) tree view showing a hierarchy of folders and scenarios, including "insert\_MPN005" and "INSERT\_MPN005 バージョン 001". A yellow callout bubble points to the scenario with the text "シナリオが生成されます" (Scenario is generated).
- Center:** A "新規シナリオ" (New Scenario) dialog box with "名前" (Name) set to "INSERT\_MPN005" and "バージョン" (Version) set to "001". Red arrows point to the "OK" button and the "名前" field.
- Bottom Left:** The "プロジェクト" tree view again, with a context menu open over the "insert\_MPN005" folder. The "シナリオの生成..." (Generate Scenario...) option is highlighted. A red arrow points from this menu to the dialog box.
- Bottom Center:** A Windows command prompt window showing the execution of the scenario:

```
C:\WINDOWS\system32\cmd.exe
D:\ODI10134\oracledi\bin>startscen insert_MPN005 001 DEV
OracleDI: Starting scenario insert_MPN005_001 DEV
D:\ODI10134\oracledi\bin>
D:\ODI10134\oracledi\bin>
D:\ODI10134\oracledi\bin>
```
- Bottom Right:** A yellow callout bubble with the text "シナリオを生成するとOSコマンドから実行できる" (Scenario can be executed from OS command after generation) and "ジョブコントローラ等の外部アプリケーションとの連携も容易" (Easy integration with external applications like job controllers).

ORACLE



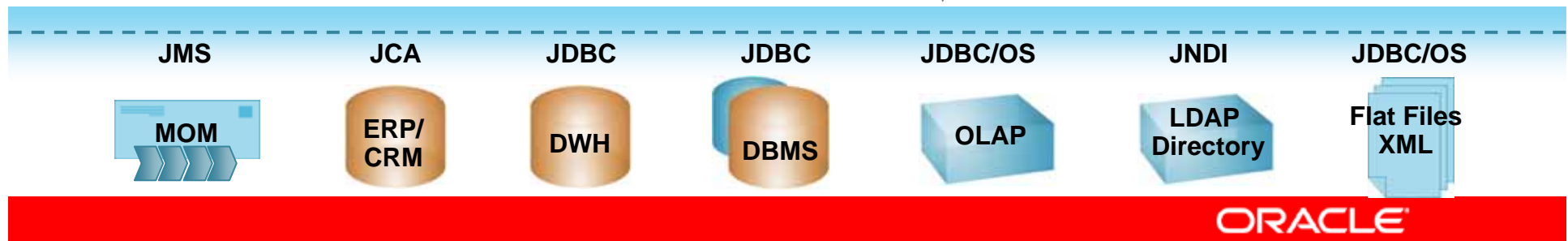
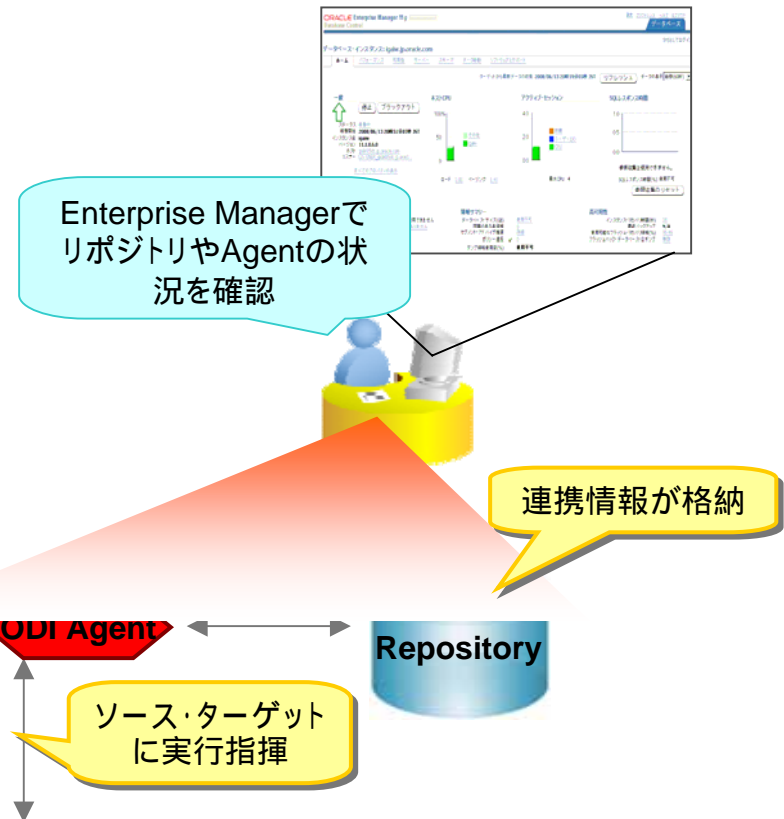
# ODI の死活監視

リポジトリ・Agentの停止を迅速に検知・対応が必要

リポジトリやAgentに問題が生じると連携を行うことが困難

リポジトリ・Agentの死活管理が必要

- リポジトリ (Oracle Database) の死活監視
  - Oracle Enterprise Manager の標準機能でリポジトリを監視
- Agent の死活監視
  - Agent のJava プロセスを監視するスクリプト (sh/bat) を作成
  - Agentを複数稼動



# ODI の死活監視

Metadata NavigatorからもAgentのステータスを確認できる

The screenshot shows the Oracle Metadata Navigator interface in a Mozilla Firefox browser window. The page title is "メタデータ・ナビゲータ" and the URL is "http://localhost:8080/oracledimn/index.jsp". The user is identified as "SUPERVISOR@[WORKREP1]".

The main content area displays a table titled "スケジュールされた実行リスト" (Scheduled Execution List). The table has five columns: "ステータス" (Status), "エージェント" (Agent), "シナリオ名" (Scenario Name), "バージョン" (Version), and "コンテキスト" (Context). Two columns, "ステータス" and "エージェント", are highlighted with red boxes. Yellow callout boxes labeled "ステータス" and "Agent名" point to these columns.

ステータス	エージェント	シナリオ名	バージョン	コンテキスト
非アクティブ	local_host	SUMMARY.001	001	Global
非アクティブ	local_host	HIROMU01SA_TO_LOCALORACLE_ORGKM.001	001	Global
非アクティブ	local_host	DATEFILE_GENERATE.001	001	Global
非アクティブ	local_host	SA_TO_ORACLE_NEWKM.001	001	Global
アクティブ	local_host	HIROMU01SA_TO_LOCALORACLE.001	001	Global
間隔に対してアクティブ	local_host	SUM_TRAILER.001	001	Global

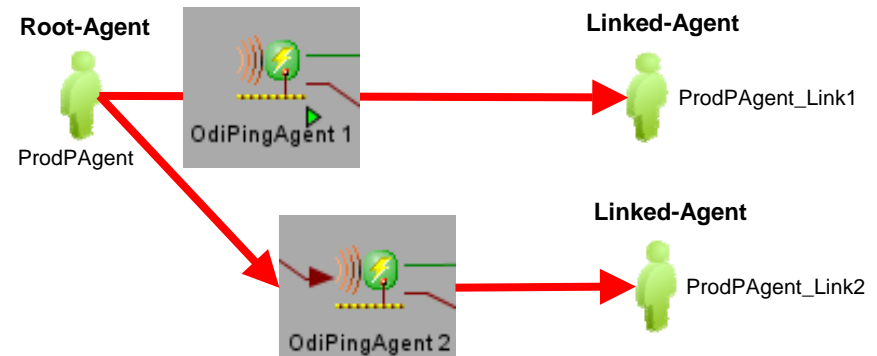
# Agentの死活監視について

Agentに親子関係をつけることで、死活監視及びフェイルオーバー、ロードバランスを行う

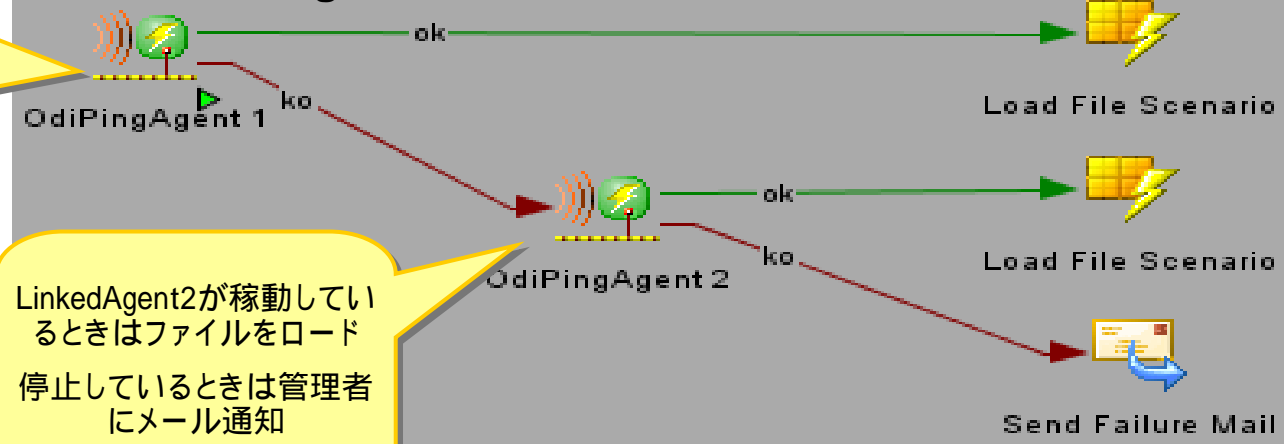
Agentに親子関係をつけることで、

- RootAgent(親Agent)がLinkedAgent(子Agent)の死活監視を行う
- LinkedAgentが停止しているとき、フェイルオーバーを行う
- LinkedAgentの負荷状況を考慮し、ロードバランスを行う

## ジョブフローとAgentの関係



## ジョブフローとAgentの関係



LinkedAgent1が稼動しているときはファイルをロード  
停止しているときはLinkedAgent2にフェイルオーバー

LinkedAgent2が稼動しているときはファイルをロード  
停止しているときは管理者にメール通知

# まとめ

Oracle Data Integratorで、様々な利用シーンにあった運用方法を柔軟に設定

## Oracle Data Integrator利用シーン

- バッチ連携
- リアルタイム連携
- SOA連携
- パッケージアプリケーション連携

## Oracle Data Integrator運用ノウハウ

- 連携構成を把握することで、構成変更時の影響を制限
- メール通知などにより迅速なエラー・障害通知
- 連携全体のジョブ監視
- ODIコンポーネント監視
- Agentに対するフェイルオーバー設定



# 年末ダイセミ受講感謝キャンペーン

Oracle Direct Seminarを御愛護頂き、誠にありがとうございます。感謝の気持ちを込めまして、**合計100名様**にWendy2010年版カレンダーをプレゼントいたします。11月・12月に開催のダイセミを2つ以上受講頂いた方が対象です。是非皆様奮ってご応募下さい!!!

プレゼントの送付先は、セミナー登録時にご登録されている貴社住所宛てに送付させていただきます。お客様の登録情報に、a.貴社名、b.部署名、c.役職名、d.住所が正しく登録されていることをご確認ください。a,b,c,dの情報が**正しく登録されていない場合はご応募が無効**となりますのでご注意ください。お客様情報の変更はこちらから実施頂けます。

<http://www.oracle.com/technology/global/jp/membership/index.html>

## 応募方法



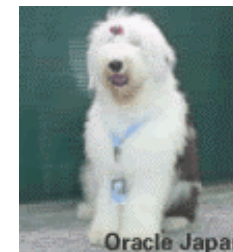
[ORD\\_SEMINAR\\_JP@ORACLE.COM](mailto:ORD_SEMINAR_JP@ORACLE.COM)

【タイトル】年末カレンダー応募

【必要情報】

- 1、ご登録の氏名
- 2、ご登録の貴社名、所属部署名
- 3、受講された2009年11月・12月開催のセミナータイトル
- 4、現在ご検討中のシステムについてなど、Oracle Directに相談されたいことなどございましたら記載ください。

必要情報を明記のうえ、メールでご応募ください。当選者の発表は発送をもってかえさせていただきます。



ORACLE

# OTN揭示版×ダイセミ でスキルアップ!!

- ・セミナー中に解消できなかった疑問点を解消したい！
- ・セミナー終了後に疑問点が出てきた！
- ・一般的なその解決方法などを知りたい！



このようなお客様に、  
Oracle Technology Network(OTN)の  
揭示版の活用をお薦めします。

<http://otn.oracle.co.jp/forum/index.jspa?categoryID=2>

→ セミナーに関連する質問については、OTN揭示版の  
「データベース一般」へ

OTN揭示版は、基本的にOracleユーザー有志からの回答となるため100%回答があるとは限りません。  
ただ、過去の履歴を見ると、質問の大多数に関してなんらかの回答が書き込まれております。

ORACLE

## Oracle University Live Virtual Class

無料ミニセッション参加で

Oracle University LVCコースが**15% OFF**！

スキルアップ  
応援キャンペーン

インターネットを通じてライブで研修に参加できる**Oracle University Live Virtual Class (Oracle University LVC)**のスキルアップ応援キャンペーンです。



その

1時間の**Oracle University LVC無料ミニセッション開催！**

オラクルユニバーシティの**人気講師**が、**定番&最新の研修コース**から気になるトピックをピックアップしてお届けします。

その

参加者全員に、期間限定で**Oracle University LVCコース15%OFF**をプレゼント！

### 無料ミニセッションお申し込み

データベース  
ミドルウェア

無料セッション  
& 割引詳細

『パフォーマンスチューニング』『Data Warehouse』などに  
ピンと来た方、必見です！

(開催日:11/2、11/12、11/13)

[http://education.oracle.co.jp/lvc\\_session\\_0911/](http://education.oracle.co.jp/lvc_session_0911/)

ビジネス・  
アプリケーション

無料セッション  
& 割引詳細

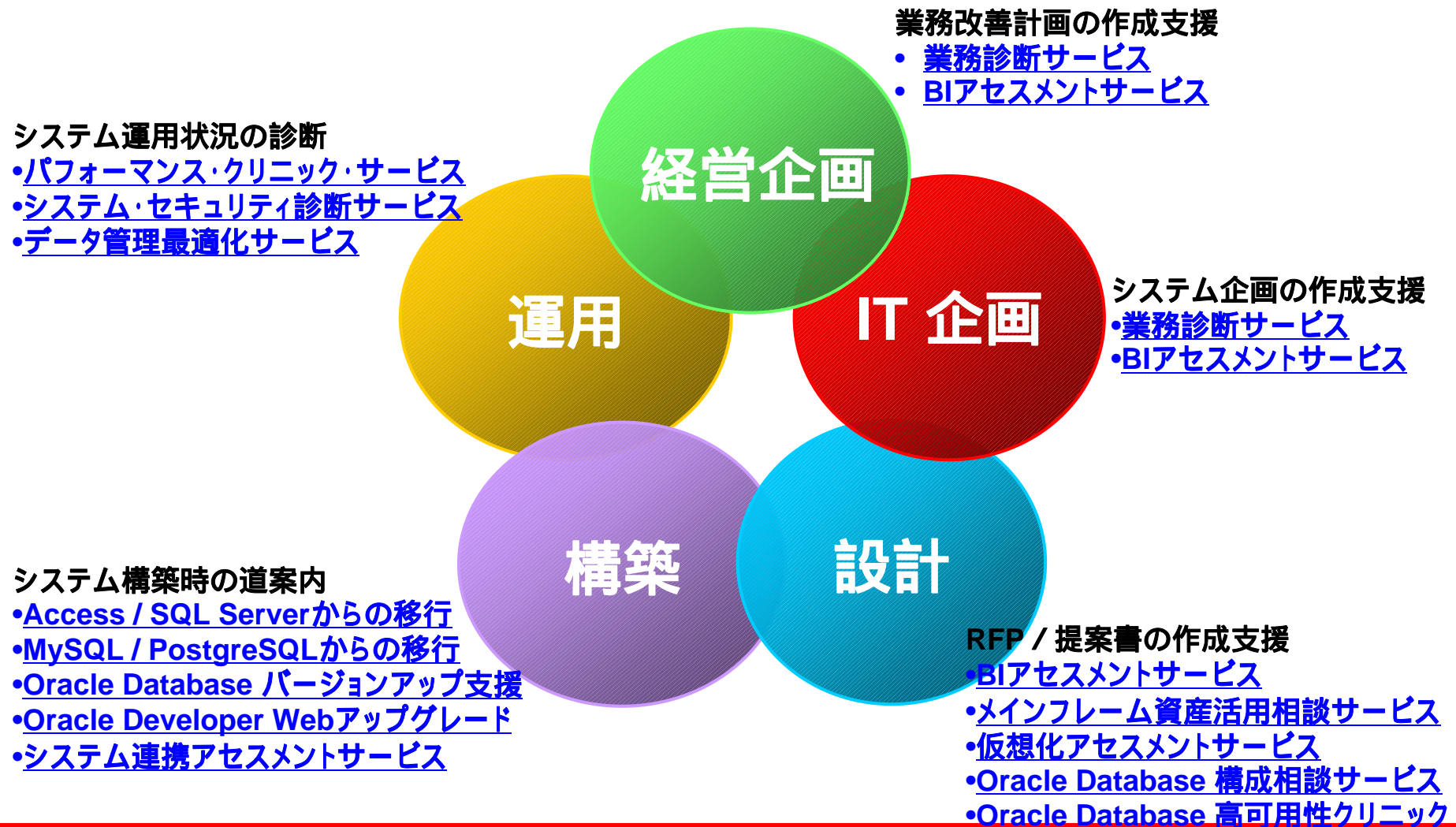
Oracle E-Business Suiteの新機能から、Siebel、PeopleSoft、  
JD Edwards、Agileの入門コースまで、気になるトピックが勢揃い！

(開催日:11/20、11/27、12/3、12/4)

[http://education.oracle.co.jp/lvc\\_session\\_0912/](http://education.oracle.co.jp/lvc_session_0912/)

# ITプロジェクト全般に渡る無償支援サービス

## Oracle Direct Conciergeサービスメニュー



ORACLE



あなたにいちばん近いオラクル



# Oracle Direct

まずはお問合せください

システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。

システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

## Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。

[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

フォームの入力には、Oracle Direct Seminar申込時と同じ  
ログインが必要となります。

こちらから詳細確認のお電話を差し上げる場合がありますので、ご登録さ  
れている連絡先が最新のものになっているか、ご確認下さい。

## フリーダイヤル

**0120 - 155 - 096**

月曜~金曜 9:00~12:00、13:00~18:00

(祝日および年末年始除く)

ORACLE



# ORACLE®

以上の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Oracle、PeopleSoft、JD Edwards、及びSiebellは、米国オラクル・コーポレーション及びその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標の可能性がります。



ORACLE®