Introducing Oracle Data Integrator and Oracle GoldenGate

Marco Ragogna
EMEA Principal Sales Consultant
Data integration Solutions
IT Obstacles to Unifying Information

What is it costing you to unify your data?

Fragmented Data Silos  
Slow Performance  
Out of sync  
Poor Data Quality  
What's the cost?
Data Integration
Key Component of Oracle Fusion Middleware

- **Applications**
- **Middleware**
- **Database**
- **Infrastructure & Management**

**Development Tools**
Unified Development Tools and Frameworks for Java EE, SOA, BPM, BI, RIA, and Security

**User Experience**
Portal, Web 2.0, Social, Collaboration, Team Spaces, RIA, Mobile, Conferencing, Presence

**Content Management**
Web Content, Document, Records Mgmt, DAM, Capture and Imaging, Archiving, IRM

**Business Intelligence**
Query & Analysis, OLAP, Dashboards, Reports, Alerts, Real-Time

**SOA & Process Management**
Process Modeling & Simulation, ESB, BPEL, Workflow, BAM, Rules, B2B, Governance, CEP

**Data Integration**
Real-Time Data Replication, ETL/E-LT, Data Quality & Profiling, Data Services, MDM

**Application Grid**
Java EE, Application Servers, Clustering, JVMs, In-Memory Data Grid, Client Java

**Identity Management**
Provisioning, Access Management, Audit, Directory, Role Management, Fraud Detection

**Enterprise Management**
Complete Middleware Management: Monitoring, Diagnostics, Configuration, Testing & Change Mgmt
Oracle Data Integration
The solution for enterprise-wide real-time data

Dramatically improve the accessibility, reliability, and quality of critical data across enterprise systems
Oracle Data Integration
The solution for enterprise-wide real-time data

Dramatically improve the accessibility, reliability, and quality of critical data across enterprise systems
Oracle Data Integration
The solution for enterprise-wide real-time data

Dramatically improve the accessibility, reliability, and quality of critical data across enterprise systems

Oracle Golden Gate
ODI EE
Enterprise Data Quality

Databases
Distributed systems
Legacy systems
OLAP systems
OLTP systems

Mission critical systems and data
Business Intelligence, Performance Management
Data Warehouses, MDM
SOA
Why Does ODI Win?

ODI is Faster
- Fastest E-LT Bulk/Batch Performance
- Faster Real Time integration (sub-second trickle) with CDC, Replication, and SOA infrastructure
- Faster Project Setup, Design and Delivery

ODI is Simpler
- Simpler Setup, Configuration, Management, and Monitoring
- Simpler way to do Mapping using Declarative SQL Interfaces
- Simpler Deployment with Fewer Hardware Devices
- Simpler extensibility with Knowledge Module code templates

ODI is Saves Money (Lower TCO, Higher ROI)
- Less Hardware & Energy Costs with E-LT Architecture
- Less Time Wasted on Unnecessary ETL Mappings, Scripting, and Complex Training
- Less Integration Overhead Integrating with Applications, SOA, and Management Software
ODI Saves Money
E-LT Runs on Existing Servers with Shared Administration

Typical: Separate ETL Server
- Proprietary ETL Engine
- Expensive Manual Parallel Tuning
- High Costs for Standalone Server

ODI: No New Servers
- **Lower Cost**: Leverage Compute Resources & Partition Workload efficiently
- **Efficient**: Exploits Database Optimizer
- **Fast**: Exploits Native Bulk Load & Other Database Interfaces
- **Scalable**: Scales as you add Processors to Source or Target
- **Manageability**: unified Enterprise Manager

Benefits
- Better Hardware Leverage
- Easier to Manage & Lower Cost
- Simple Tuning & Linear Scalability
ODI is Faster
Up to 7TB per hour of real world data loading and complex transformations

ODI ELT (on Exadata/any DW)
- ODI scales with the Database
  - Loads increase linearly as DW scales
- ODI runs on relational technologies – no ETL hardware required
  - No new hardware required as data sets grow
  - ODI processes used only during integration runs
  - Databases continually available for OLTP, BI, DW, etc
- Common administration, monitoring and management
- All the benefits of rapid tools-based ETL development

Conventional ETL
- As data sets grow, more hardware ($$) needed to scale
- ETL parallel optimization and design ($$$$) is heavily dependent on resources available to the ETL environment
  - Sources, integrations, targets must be designed to match processing power of ETL environment
  - Source flat files split to match # of ETL engine CPU’s
  - Integration grid setup appropriately to match # of ETL engine CPU’s
  - Target partitions, table spaces to match # of ETL engine CPU’s
- ETL engine hardware resources only used for ETL
  - Cannot be utilized for OLTP, BI, DW, etc.
- Hardware not co located, multiple vendors
- Different management, monitoring and administration from database and BI infrastructure ($$)
“Old Style” ETL

- Monolithic & Expensive Environments
- Fragile, Hard to Manage
- Difficult to Tune or Optimize

ETL engines require BIG H/W and heavy parallel tuning

Monolithic data streaming architecture
Modern Data Integration

- Lightweight, Inexpensive Environments – Agents
- Resilient, Easy to Manage – Non-Invasive
- Easy to Optimize and Tune – uses DBMS power

Extract       Transform            Load       Lookups/Calcs

Set-based SQL transforms typically faster

Flexible options for real time data streams

SQL Load inside DB is always faster
Best Data Integration for Exadata
Top Performance, Smallest Footprint

• Run ODI, EDQ & OGG Directly on Exadata
• Support Any Latency Data Feeds
• Non-Invasive Source Capture
• Most Cost-Effective and High-Performance Exadata Data Loading

Oracle GoldenGate
Oracle Data Integrator
Oracle Enterprise DQ

Non-Invasive Real Time Transaction Feeds

Batch Feeds, Incremental Updates and in-DB transformations via ELT
ODI is Simpler

Speed Project Delivery and Time to Market with ODI

• Development Productivity
  • 40% Efficiency Gains

• Environment Setup (ex: BI Apps)
  • 33-50% Less Complex

**ODI Declarative Design**

1. Define *What* You Want
2. Automatically Generate Dataflow

Define *How*: Built-in Templates

**Conventional ETL Design**

- Number of Setup Steps: 7
- Number of Servers: 1
- Number of connections: 3

- Number of Setup Steps: 10
- Number of Servers: 3
- Number of connections: 7
One or a related group of flow-based procedural ETL Mappings – first sample

One declarative ODI interface plus selection among existing Knowledge Modules

One or a related group of flow-based procedural ETL Mappings - second sample
Traditional procedural ETL

Traditional ETL row to row complexity

Flow Generation is AUTOMATIC, written by ODI directly!
You describe how the relational infrastructure where ODI works is done
- ODI builds the flow for a specific loading automatically!

Topology module allows to describe all the information on the technology where
the ELT projects work, starting from specific definition on the technologies that
are used, going to physical description on how to access a server, which user
and password to enter, which schema users or database are involved in the
jobs. The final developer will have only a logical reference to the servers
Declarative mapping + Knowledge Modules = Generated Code

**KM’s Meta Code**

```sql
INSERT INTO <%= smnRef.getTable() %>
VALUES (<%= smnRef.getPk() %>);
```

**Metadata**

**Pluggable Knowledge Modules Architecture**

- **Reverse**
  - Engineer Metadata
  - Read from CDC Source

- **Journalize**
  - From Sources to Staging

- **Load**
  - Constraints before Load
  - Transform and Move to Targets

- **Check**
  - Staging Tables
  - Target Tables
  - Error Tables

- **Integrate**

- **Service**
  - Expose Data and Transformation Services

120+ KMIs out-of-the-box
- Tailor to existing best practices
- Ease administration work
- Reduce cost of ownership

Customizable and extensible

- **Declarative mapping + Knowledge Modules = Generated Code**

- **Tailor to existing best practices**
- **Ease administration work**
- **Reduce cost of ownership**

- **Customizable and extensible**

**Pluggable Knowledge Modules Architecture**

- **KM Interpreter**
- **Executed Code**

- **Metadata**
- **KM's Meta Code**

- **Pluggable Knowledge Modules Architecture**

- **Reverse**
  - Engineer Metadata
  - Read from CDC Source

- **Journalize**
  - From Sources to Staging

- **Load**
  - Constraints before Load
  - Transform and Move to Targets

- **Check**
  - Staging Tables
  - Target Tables
  - Error Tables

- **Integrate**

- **Service**
  - Expose Data and Transformation Services

**KM Interpreter**

**Executed Code**

```sql
INSERT INTO db_staging.Region_AGGREGATED_SAK
VALUES (REGION_NAME, COUNTRY_NAME, CELL_NAME);
```
Jobs, auditing

- Technical and business metadata: ability to manage in a unique and centralized way jobs, their transformation, schedulings, data definition language etc.
- Central Monitoring and Logging: verifying the execution of jobs

Graphical environment allows to describe job complex as needed, created putting together simple steps like the declarative design

ELT Agent writes back on the repository the auditing offor the job executions, giving information on generated code, warnings and database errors that can eventually occur
Oracle Data Integration
The solution for enterprise-wide real-time data

Dramatically improve the accessibility, reliability, and quality of critical data across enterprise systems
Oracle GoldenGate Overview

**Oracle GoldenGate provides low-impact capture, routing, transformation, and delivery of transactional data across heterogeneous environments in real time**

Key Differentiators:

- **Performance**: Non-intrusive, low-impact, sub-second latency
- **Flexible and Extensible**: Open, modular architecture - Supports heterogeneous sources and targets
- **Reliable**: Maintains transactional integrity - Resilient against interruptions and failures
Oracle GoldenGate Use Cases
Enterprise-wide Solution for Real Time Data Needs

Log Based, Real-Time Change Data Capture

Zero Downtime Migration and Upgrades
New DB/OS/HW/App

Active-Active High Availability
Fully Active Distributed Database

Query Offloading
Reporting Database

Real-time BI
ETL
EDW

Data Distribution
Global Data Centers

Heterogeneous Source Systems

ETL
ODS
EDW

SOA/EDA

Reduce Costs
Lower Risks
Achieve Operational Excellence
Advantages of Oracle GoldenGate Architecture

<table>
<thead>
<tr>
<th>Reduced Overhead and TCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Captures once, delivers to many targets for different uses</td>
</tr>
<tr>
<td>• Non-invasive, log-based capture</td>
</tr>
<tr>
<td>• Moves only committed data, reduces bandwidth needs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>High Performance with Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Subsecond latency even with high data volumes</td>
</tr>
<tr>
<td>• Preserves transaction integrity</td>
</tr>
<tr>
<td>• Ensures data recoverability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flexibility and Ease of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Provides decoupled, modular architecture</td>
</tr>
<tr>
<td>• Supports heterogeneous sources and targets, and different latency needs</td>
</tr>
<tr>
<td>• Coexists and integrates with ELT/ETL and messaging solutions</td>
</tr>
</tbody>
</table>
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

Source Oracle & Non-Oracle Database(s)

LAN/WAN Internet

Target Oracle & Non-Oracle Database(s)
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

**Trail**: stages and queues data for routing.
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

**Trail**: stages and queues data for routing.

**Pump**: distributes data for routing to target(s).

Capture: Source

Trail: LAN/WAN

Pump: Internet

Target: Target
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

**Trail**: stages and queues data for routing.

**Pump**: distributes data for routing to target(s).

**Route**: data is compressed, encrypted for routing to target(s).

Source
Oracle & Non-Oracle Database(s)

Capture
Trail
Pump
LAN/WAN Internet
TCP/IP
Tail
Target
Oracle & Non-Oracle Database(s)
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

**Trail**: stages and queues data for routing.

**Pump**: distributes data for routing to target(s).

**Route**: data is compressed, encrypted for routing to target(s).

**Delivery**: applies data with transaction integrity, transforming the data as required.
How Oracle GoldenGate Works

**Capture**: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

**Trail**: stages and queues data for routing.

**Pump**: distributes data for routing to target(s).

**Route**: data is compressed, encrypted for routing to target(s).

**Delivery**: applies data with transaction integrity, transforming the data as required.

Bi-directional

Source
Oracle & Non-Oracle Database(s)

LAN/WAN
Internet

TCP/IP

Target
Oracle & Non-Oracle Database(s)
GoldenGate Checkpointing

- Capture, Pump, and Delivery save positions to a checkpoint file so they can recover in case of failure.
Zero Downtime Oracle Upgrade Implementation Steps: Example of 9i → 11g Cross-Platform

1. Start Oracle GoldenGate Capture module
2. - 4. Initial loading, export import of a new 11g target db (ELT/flat files/jdbc/native db loaders/import export tablespaces etc.)
5. Start Oracle GoldenGate Delivery module at target
6. Start Oracle GoldenGate’s Capture at 11g
7. Start Oracle GoldenGate’s Delivery process 9i (old source, contingency)
# Oracle GoldenGate 11g: Heterogeneity

## Databases

**Oracle GoldenGate Capture:**
- Oracle
- DB2 for v 9.7
- Microsoft SQL Server
- Sybase ASE
- Teradata
- Enscribe
- SQL/MP
- SQL/MX
- MySQL
- JMS message queues

**Oracle GoldenGate Delivery:**
- All listed above, plus:
  - TimesTen, DB2 for iSeries
  - Exadata, Netezza, Greenplum, and HP Neoview

## O/S and Platforms

- Linux
- Sun Solaris
- Windows 2000, 2003, XP
- HP NonStop
- HP-UX
- HP OpenVMS
- IBM AIX
- IBM z Series
- zLinux
Customer Example: Zero Downtime Migration
eDialog

Goals
- 24x7x365 provider of advanced e-mail and multichannel marketing solutions to business worldwide helping marketers transform conversations into conversions.
- Ensure absolute business continuity when migrating data to a new data infrastructure

Solution
- Oracle Exadata as the foundation for new data infrastructure that ensures continuous high-performance marketing services and campaign analysis.
- Used GoldenGate for a phased migration with more than 12 terabytes of data from heterogeneous legacy environments

Return on Investment
- Completed the phased migration in six months
- Gained the ability to complete the migration in phases, enabling e-Dialog to test the new environment over time
- Reduced downtime during the massive migration effort
- Improved throughput by 50% and cut report generation time in half
Customer Example: Real-Time DW on Exadata

AVEA

Goals

• Supporting campaigns management with timely customer information
• Reducing batch windows while data increases and improving the performance of ETL and reporting

Solution

• GoldenGate feeds real-time data from CRM, Billing and other key systems to ODS
• ODI extracts from the ODS and loads near real-time data to Exadata DW
• New solution replaced IBM InfoSphere Data Stage
• OBI EE is used for real-time reporting

Return on Investment

• Access to timely data for customer segmentation in the Siebel CRM campaign management system
• Batch window for the DW decreased by 50%
• Number of reports generated from the DW has increased by 10 times