

ORACLE DATA INTEGRATOR

Soluzioni per l'integrazione e
l'armonia del dato all'interno di una
organizzazione

24/10/2024

Bregata Luca

l.bregata@mediamenteconsulting.it



Project Manager & BI Senior Analyst
BU Data Science Var Group

Oracle Certifications:

- **Oracle Data Integrator 12c**
- **Oracle APEX Cloud Developer**

Principali Settori:

- **Fashion**
- **Oil**
- **Pharma**
- **Energia**



Bregata Luca

l.bregata@mediamenteconsulting.it



AGENDA

- . *Introduzione a Oracle Data Integrator*
- . *Architettura di Odi*
- . *Armonizzazione e valorizzazione del dato*
- . *Use Cases*
- . *Cloud e Big Data*
- . *Conclusioni*



Introduzione a Oracle Data Integrator



Introduzione a Oracle Data Integrator (ODI)

Oracle Data Integrator (ODI) è una piattaforma di integrazione dati progettata per costruire, gestire e automatizzare processi di estrazione, trasformazione e caricamento dei dati (ETL).

- Può integrarsi con diverse piattaforme utilizzando connettori nativi per garantire una gestione efficiente dei flussi di dati.
- Garantisce automatizzazione e orchestrazione di processi complessi, con controlli di qualità dei dati integrati.
- Facilità di utilizzo grazie all'Interfaccia grafica user friendly
- La sua architettura permette di gestire la trasformazione e integrazione dei dati in modo scalabile.
- Architettura modulare

➤ Gestione di grandi volumi di dati

Approccio E-LT

ODI Utilizza un'architettura unica e innovativa basata su un approccio E-LT (Extract, Load, Transform).

- *Extract: Estrae il dato dalla sorgente.*
- *Load: Carica il dato nell'area di staging del db di destinazione.*
- *Transform: Trasforma il dato direttamente nel database di destinazione tramite query SQL, sfruttando la potenza di calcolo nativa dei database relazionali.*

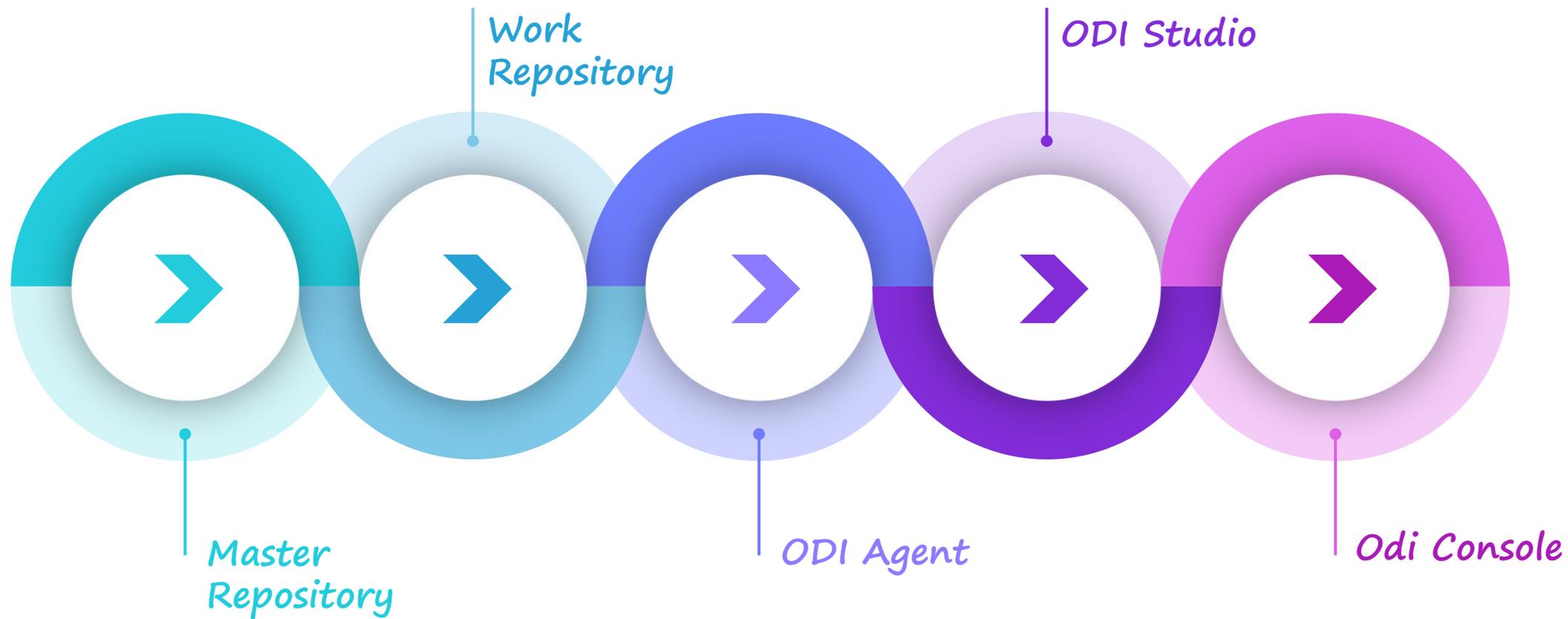
Questo approccio lo differenzia dagli strumenti ETL tradizionali, poiché ODI sfrutta al massimo le capacità e la potenza dei database esistenti di destinazione per eseguire le trasformazioni, evitando il bisogno di hardware ETL separato.

Riduce il carico sulle macchine e ottimizza l'efficienza delle risorse.

Architettura di ODI



Architettura di ODI



Master Repository

Il Master Repository centralizza la gestione delle configurazioni e delle definizioni globali, come il collegamento tra agenti e work repository, sicurezza e versionamento.

Funzioni principali:

- Gestione utenti e permessi
- Configurazione globale del sistema ODI
- Versionamento di oggetti di sviluppo
- Configurazione degli agenti
- Gestione delle connessioni



Work Repository

Contiene i dati di sviluppo e operativi, come mapping e procedure, organizzato per progetti

Funzioni principali:

- *Memorizza gli oggetti di sviluppo come modelli, mappings, scenari, procedure, pacchetti.*
- *Contiene anche i dati delle esecuzioni per monitoraggio e audit.*
- *Si possono avere diversi Work Repository per diversi ambienti (sviluppo, test, produzione).*



ODI Agent

Esegue i processi in background e permette la schedulazione.

Agente Fisico: Collegato direttamente ad un server

Agente Logico: Piattaforma virtuale per la gestione degli agenti fisici, utile per carichi distribuiti o failover e invia notifiche in caso di errori o successi.

- Standalone Agent: Agente indipendente, eseguito su un server dedicato. Gestisce l'esecuzione dei job e la schedulazione. Ideale per ambienti distribuiti.
- J2EE Agent: Eseguito su un server applicativo Java EE (come WebLogic o Tomcat). Supporta alta disponibilità e bilanciamento del carico. Perfetto



ODI Studio

Interfaccia grafica per sviluppatori e amministratori, con una vista centralizzata per gestire l'intero ciclo di vita del progetto.

Componenti principali:

- Designer: Permette di creare mapping, pacchetti, scenari e procedure.
- Operator: Utilizzato per monitorare le esecuzioni in tempo reale e gestire i job.
- Topology: Gestisce la configurazione delle sorgenti dati e delle destinazioni.
- Security: Permette di gestire ruoli e utenti.



ODI Console

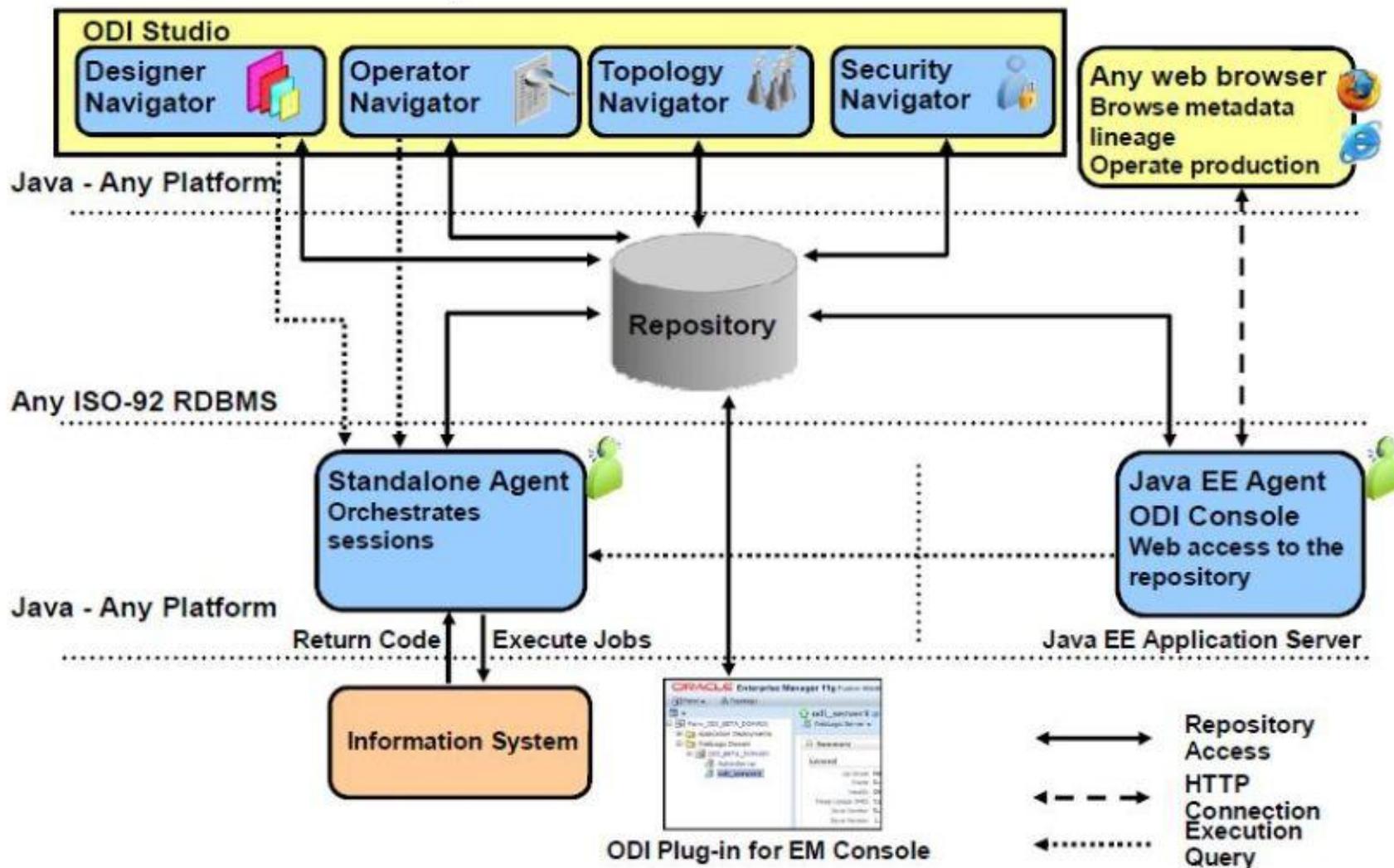
La console ODI è un'interfaccia web-based che permette il monitoraggio e la gestione delle esecuzioni di job.

Funzionalità:

- Vedere lo stato dei job in tempo reale.
- Consultare i log delle esecuzioni.
- Analizzare le performance e il consumo delle risorse.



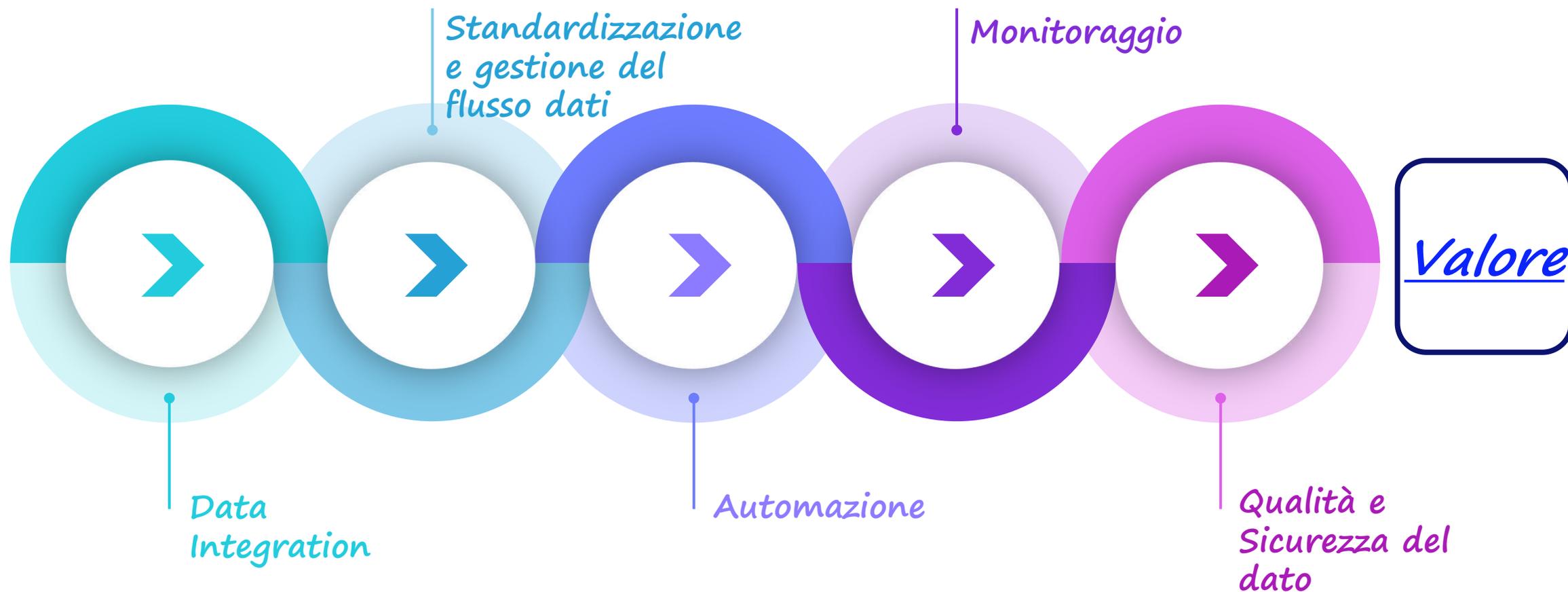
Architettura di ODI



*Armonizzazione
e e
valorizzazione
del dato*



Armonizzazione del dato



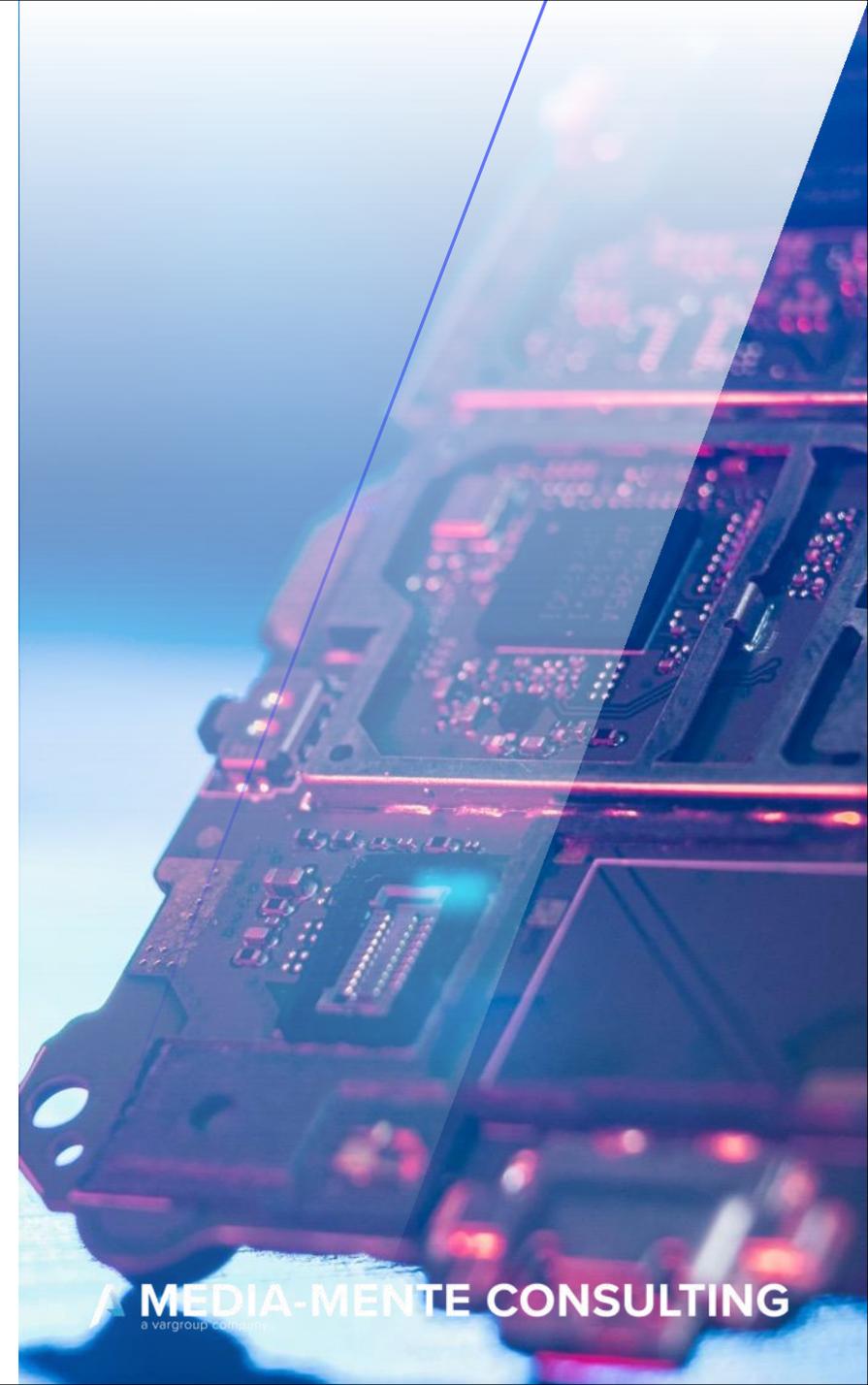
Data Integration

ODI può connettersi e integrare dati da una vasta gamma di fonti eterogenee, tra cui:

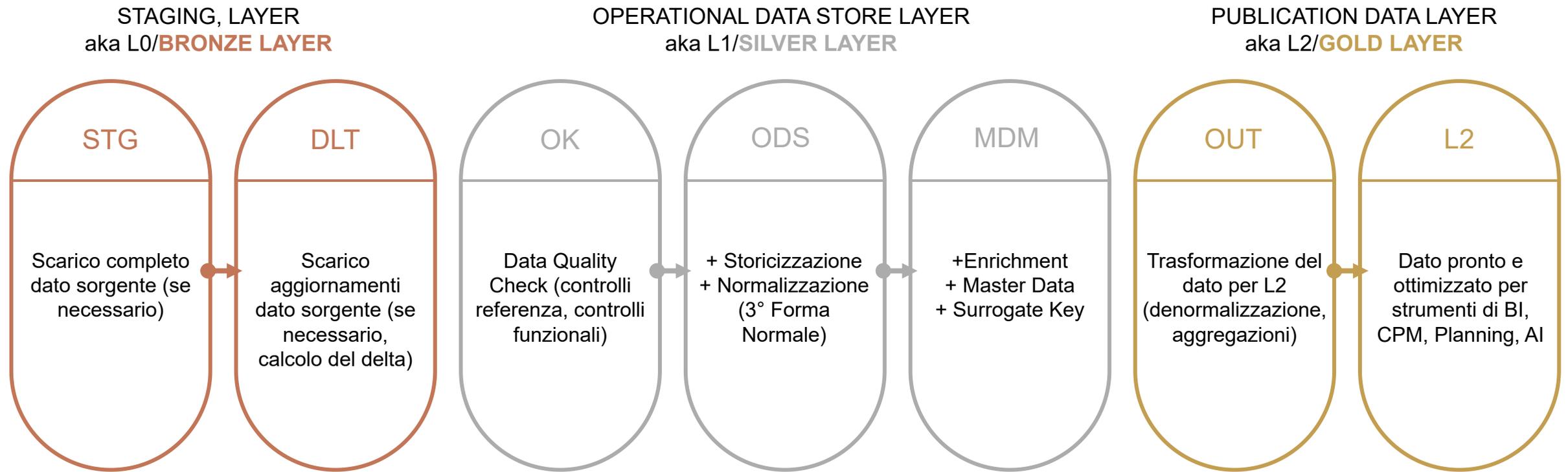
- Database relazionali: Oracle, SQL Server, MySQL...
- Flat files: CSV, XML, Excel.
- Web services e API.
- Big Data e NoSQL: Hadoop, Cassandra, MongoDB.

Funzioni principali:

- Supporto nativo per le connessioni JDBC e ODBC.
- Gestione di sorgenti dati complesse e multi-formato.
- Integrazione e trasformazione dei dati in tempo reale.



Framework ODI



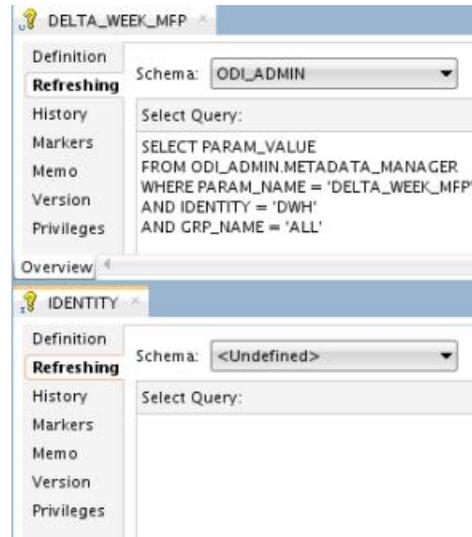
Variabili

Utilizzate per gestire valori statici o dinamici all'interno dei processi ETL quali mapping, procedure, package. I valori possono essere modificati durante l'esecuzione.

- Passaggio di valori tra steps
- Filtri e Condizioni: Le variabili possono essere utilizzate nei mapping per applicare filtri dinamici sui dati.
- Condizioni di Esecuzione: Utilizzate per controllare il flusso logico nei package o nei load plan (es. eseguire certi step solo se una variabile soddisfa una condizione).

Benefici:

- Flessibilità
- Automazione
- Riutilizzo



Procedure

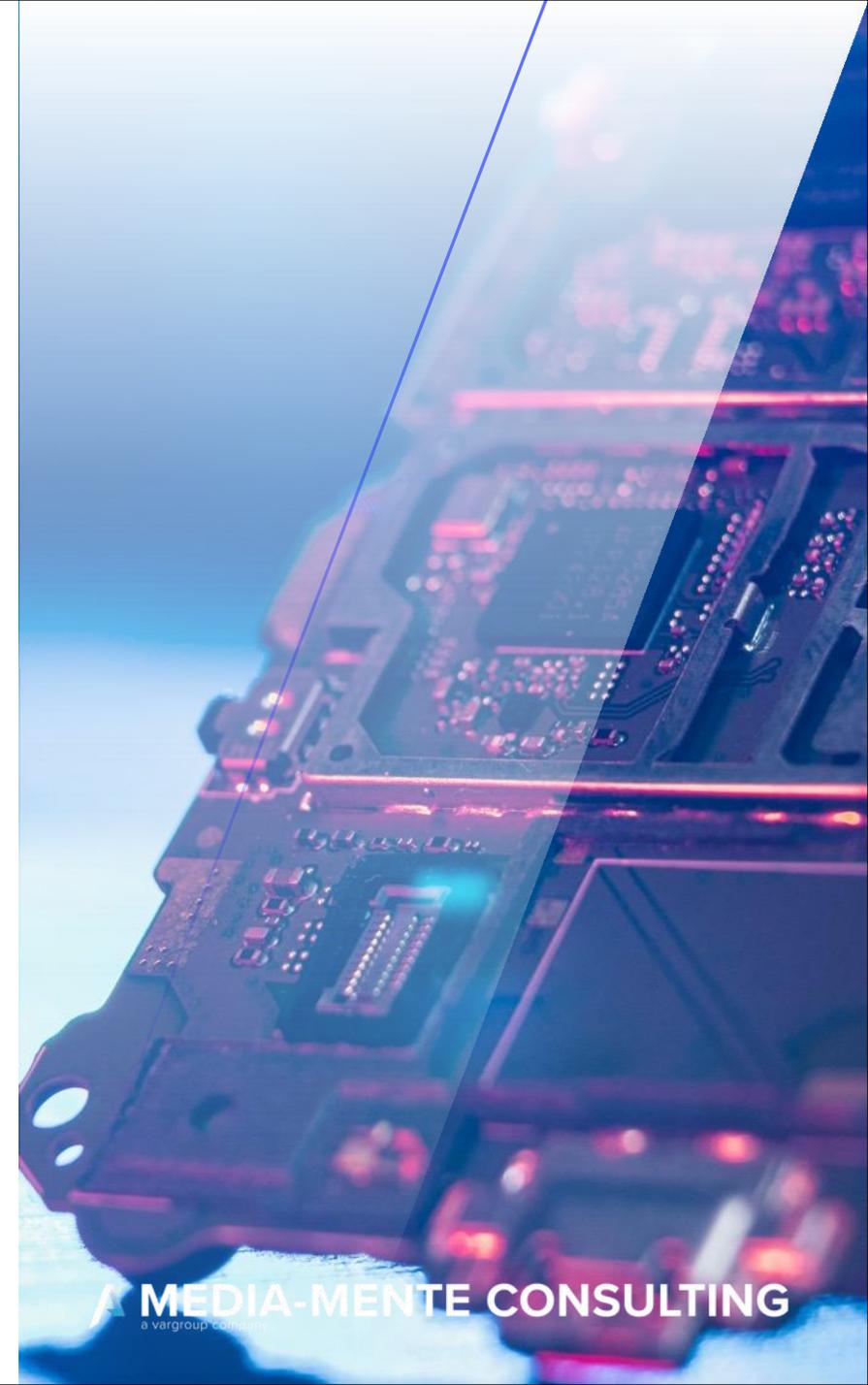
Sequenze di step predefiniti che possono includere SQL, PL/SQL, shell script o comandi di sistema.

- *Automazione di task complessi.*
- *Esecuzione di operazioni amministrative come backup, manutenzione o integrazione di dati esterni.*
- *Utilizzo per task ripetitivi e custom*

Mapping

Definisce come i dati devono essere trasformati e caricati dalla sorgente alla destinazione.

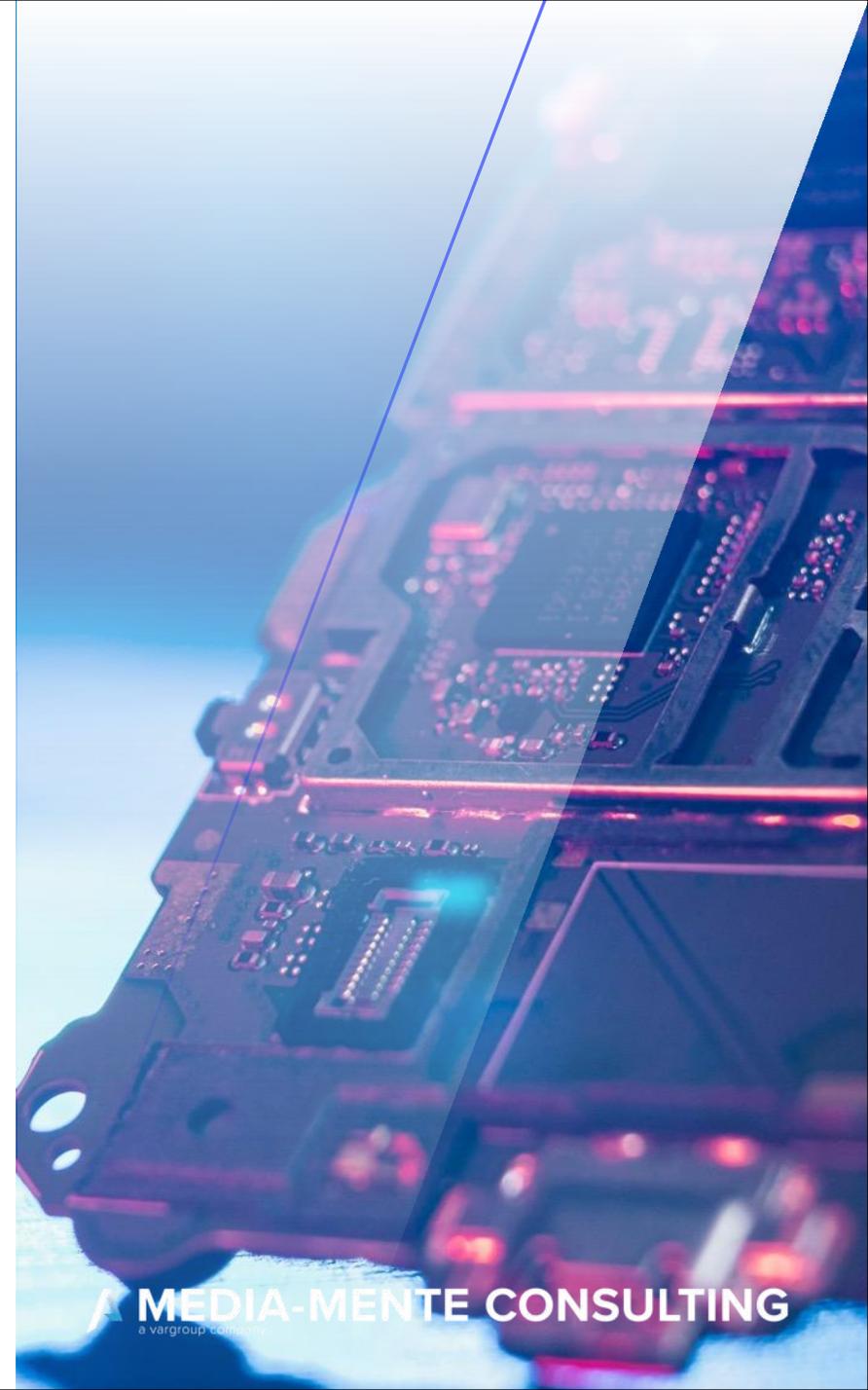
- *Definizione di regole di trasformazione (join, filtri, ecc.).*
- *Utilizzo di componenti come dataset e lookup per gestire sorgenti complesse*



Knowledge Modules (KMs)

I Knowledge Modules (KMs) sono script predefiniti che eseguono specifici compiti di integrazione (caricamento, trasformazione, ecc.)

- **LKM (Loading Knowledge Module):** Carica i dati dalla sorgente al staging con supporto per vari protocolli e formati (JDBC, File, Web Services).
- **IKM (Integration Knowledge Module):** Gestione di operazioni di inserimento, aggiornamento, eliminazione e merge ed è personalizzabile. Inoltre permette una ottimizzazione delle performance utilizzando capacità native del database (es. SQL ottimizzato).
- **JKM (Journalizing Knowledge Module):** Traccia le modifiche nelle sorgenti dati (CDC - Change Data Capture) nelle journal tables per tenere traccia delle modifiche e garantire consistenza.
- **CKM (Check Knowledge Module):** Garantisce l'integrità dei dati durante il caricamento, verificando regole come chiavi primarie e formati. Registra gli errori o interrompe il processo se i dati non superano i controlli, assicurando la qualità.



Mapping

IKM

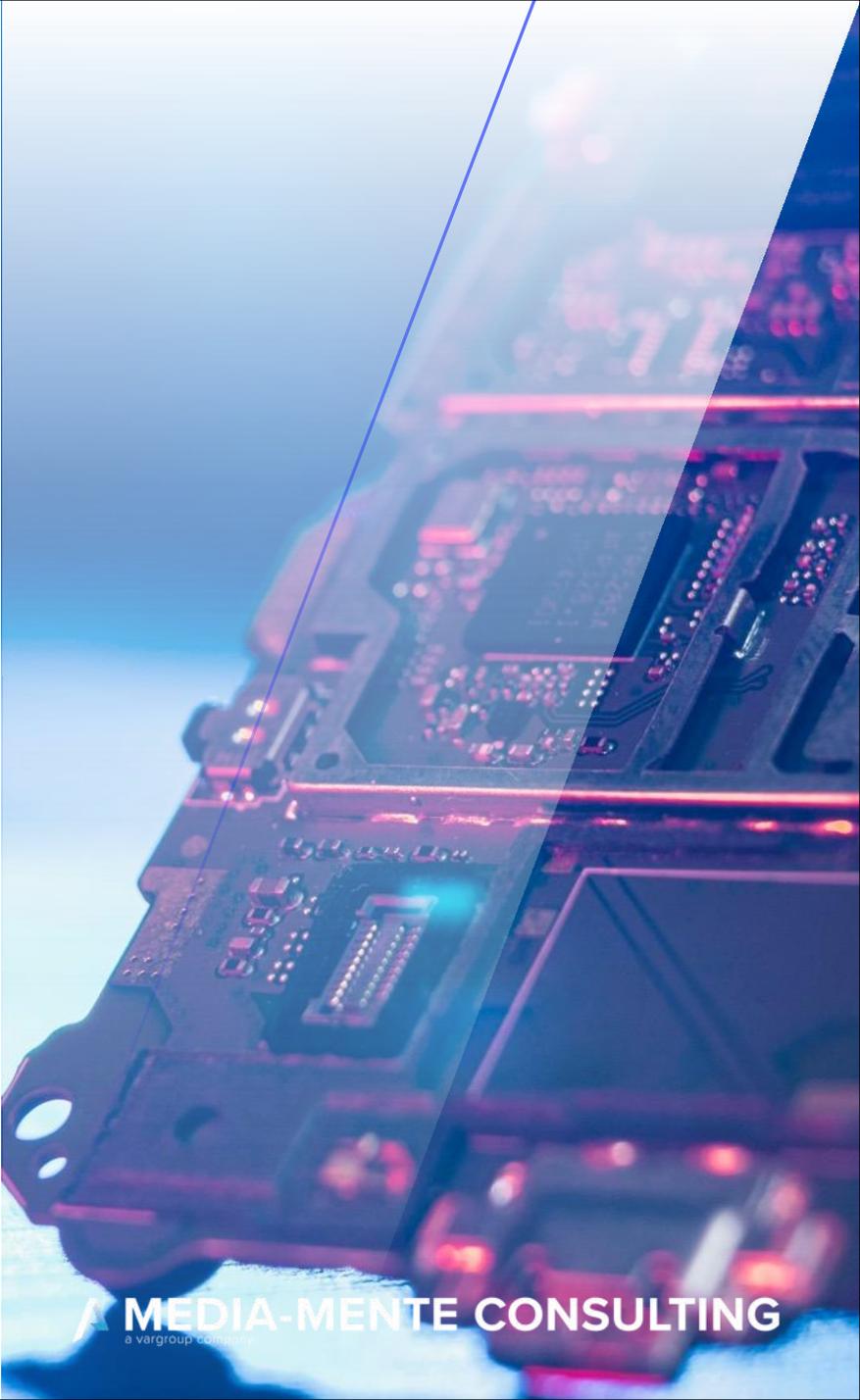
Integration Knowledge Module

Integration Knowledge Module: IKM Oracle Merge.GLOBAL

Style: Component-style

Options Description

Name	Use Default	Value	Description	Help
CREATE_TARGET_TABLE	<input checked="" type="checkbox"/>	False	Create the target table	?
MERGE_HINT	<input checked="" type="checkbox"/>		Oracle DML loading hint for loading integration table	?
SELECT_HINT	<input checked="" type="checkbox"/>		Oracle extraction hint for querying source tables	?
CONSTRAINTS	<input checked="" type="checkbox"/>	None	Disable/enable constraints before/after the data is loaded	?
TRUNCATE_TARGET_TABLE	<input checked="" type="checkbox"/>	False	Truncate the target table	?
DELETE_ALL	<input checked="" type="checkbox"/>	False	Delete all rows	?
INDEXES	<input checked="" type="checkbox"/>	None	Disable/enable indexes before/after the data is loaded in...	?

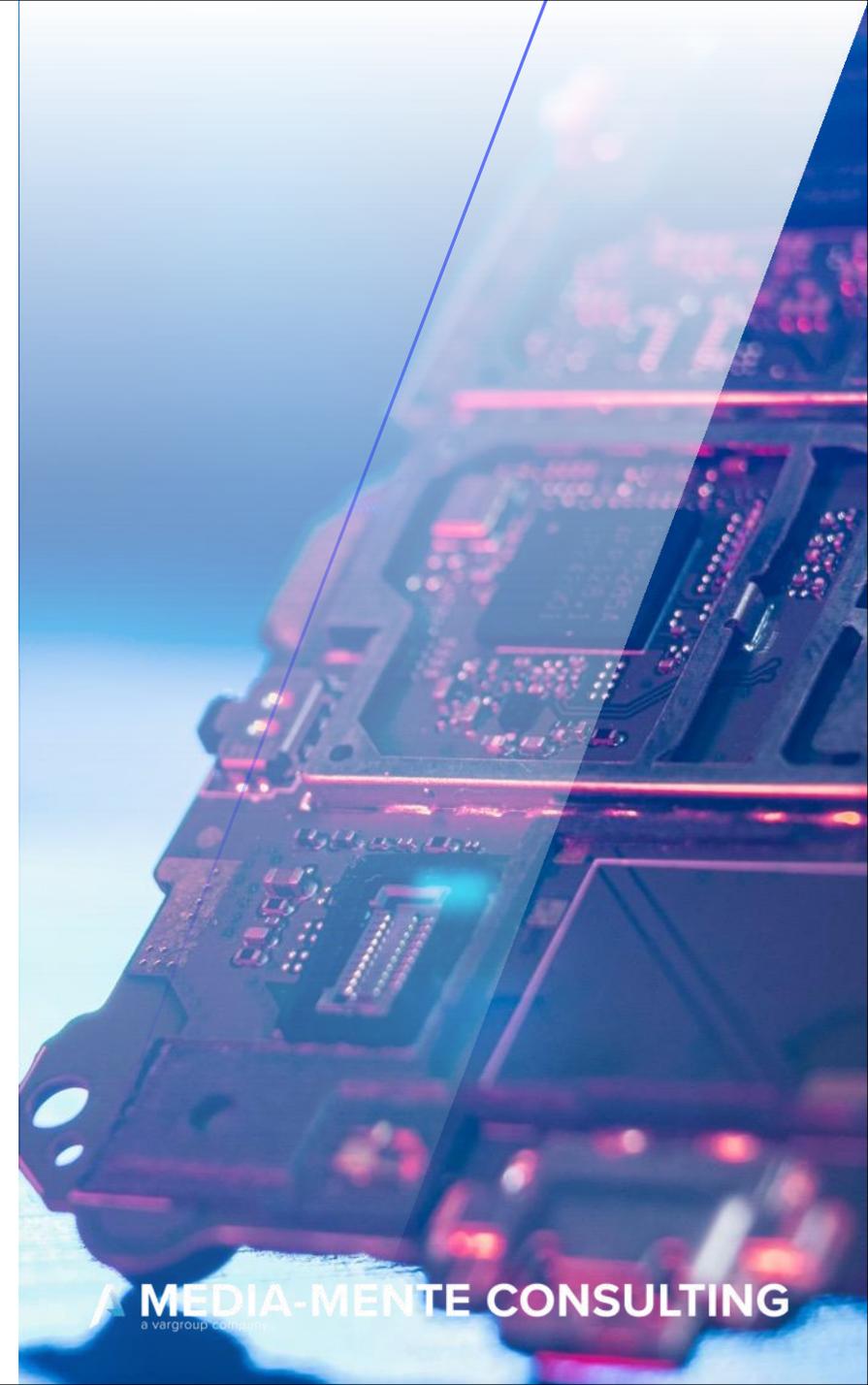


Package

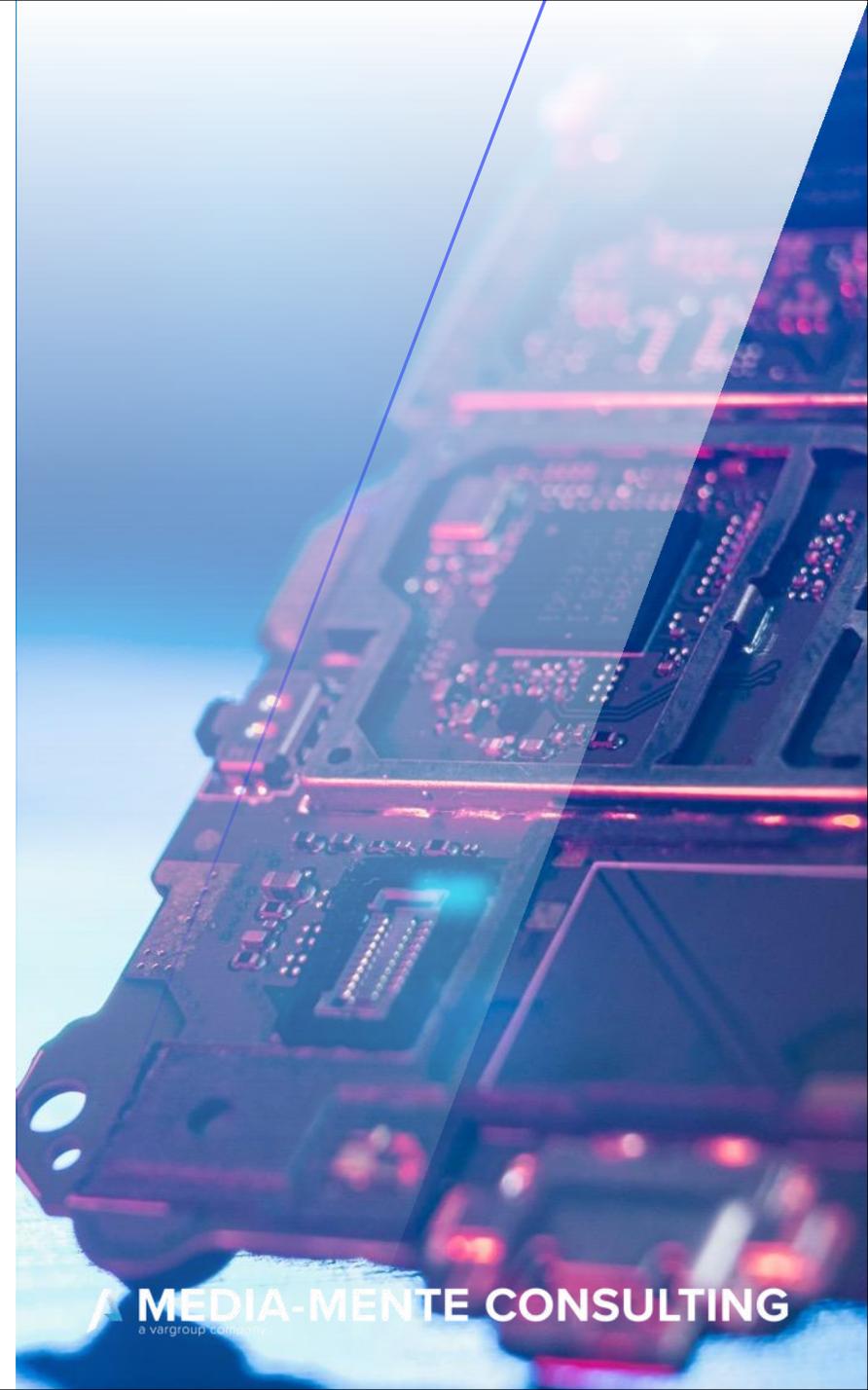
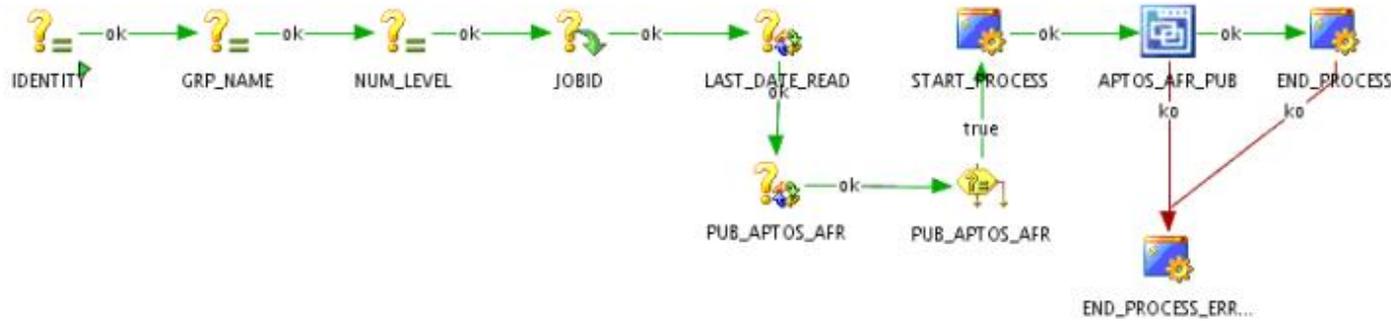
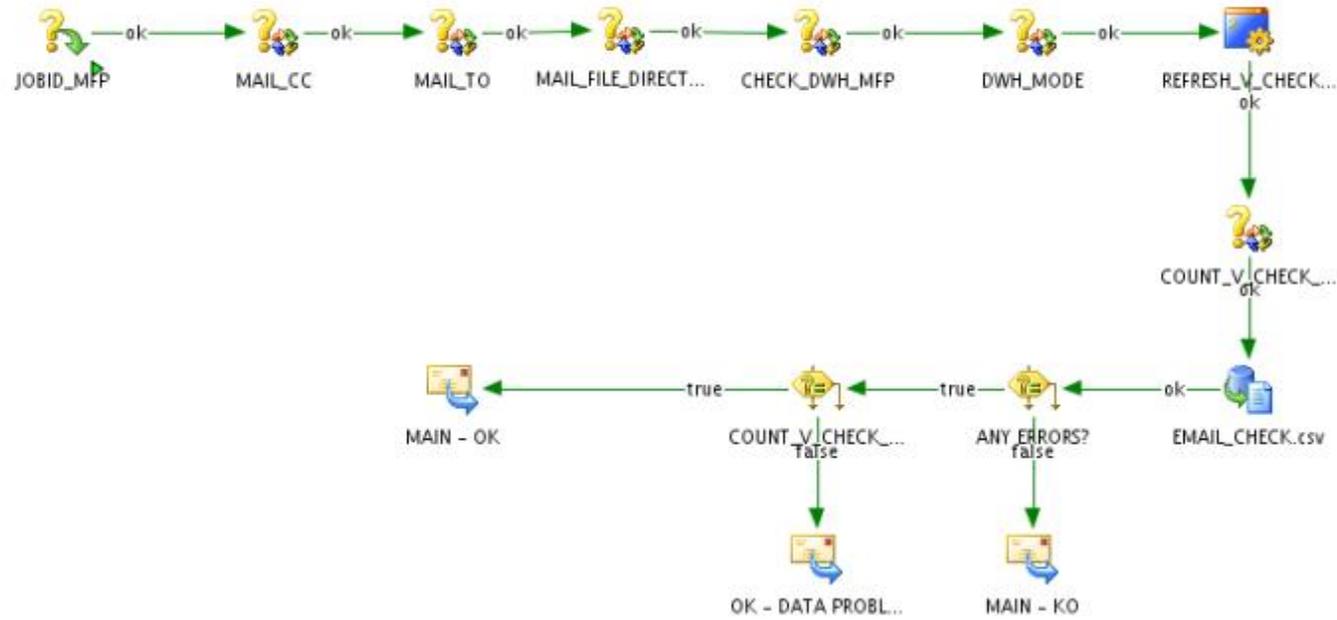
Un Package in ODI è un workflow grafico che raggruppa una serie di mapping, procedure, variabili e altri componenti in un unico flusso di lavoro.

Funzionalità:

- Consente la creazione di flussi ETL complessi che combinano diverse operazioni.
- Controllo di flusso tra step.
- Possibilità di gestire condizioni, loop, eventi.
- Utilizzo di variabili per personalizzare l'esecuzione.
- Supporta condizioni, cicli e ramificazioni per implementare logiche complesse.
- Un Package può eseguire più mapping in sequenza o in parallelo, effettuare controlli e decidere quali passaggi successivi eseguire in base ai risultati.



Package



Automazione dei flussi di lavoro

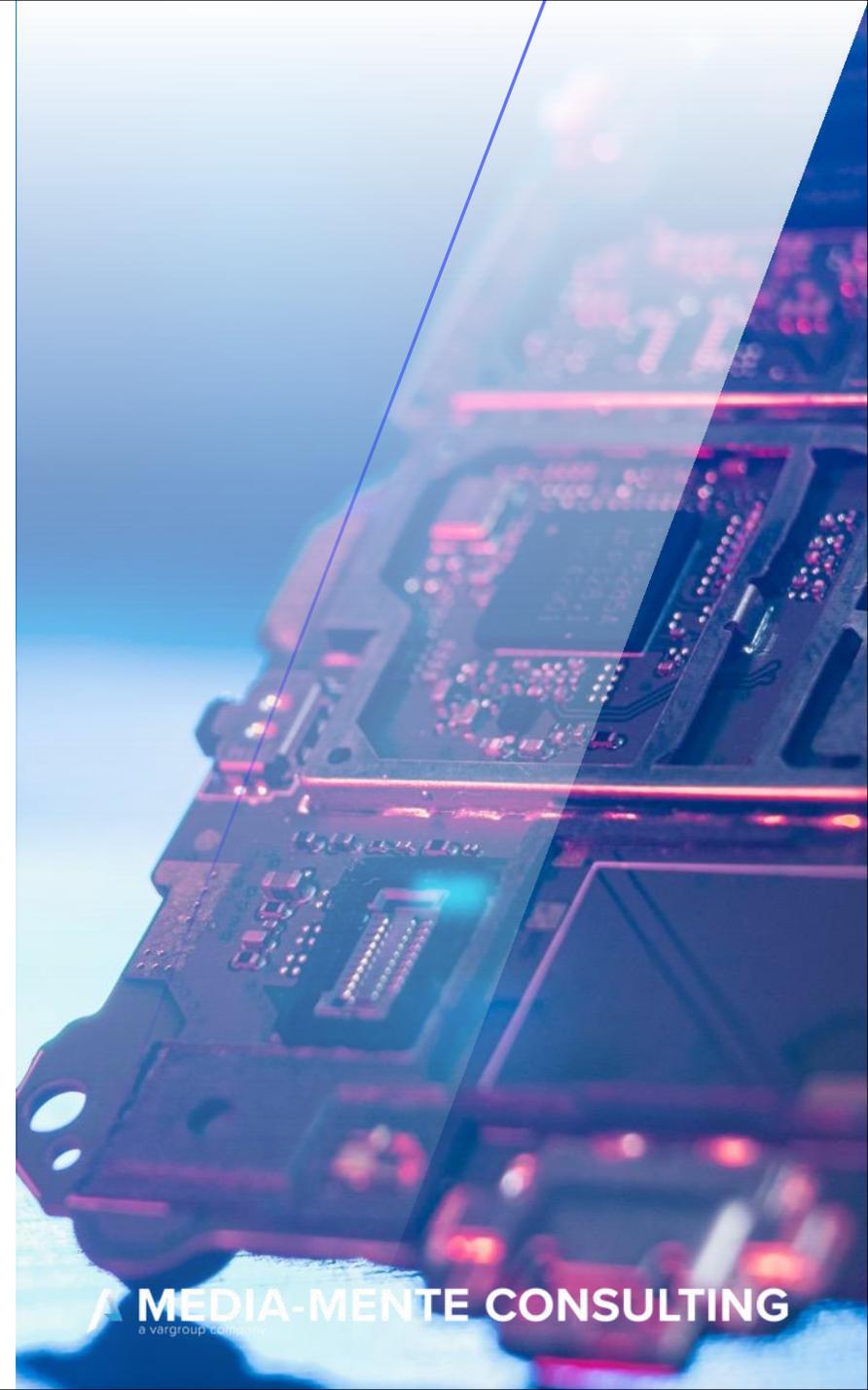
La schedulazione in ODI permette di eseguire job ETL a intervalli regolari o in base a eventi definiti, sfruttando ODI Agents che eseguono i job in background e supportano la distribuzione e l'esecuzione parallela di job su diversi server.

Funzionalità di Schedulazione:

- Pianificazione regolare per esecuzioni periodiche.
- Esecuzione basata su trigger di eventi.
- Gestione automatizzata dei job, inclusa la ripetizione in caso di errori.

Benefici:

- Riduzione dell'intervento manuale.
- Automazione di processi ETL ricorrenti.
- Gestione del carico di lavoro grazie alla distribuzione degli agenti.
- Notifiche via email in caso di successo o fallimento automatiche



Schedulazione dei Job

Load Plan Scheduling GLOBAL / odi_agent

Definition

Execution Cycle: **Scheduling [Load Plan: ODI_DWH_MAIN]**

Context: **Global** Logical Agent: **odi_agent**

Log Level: **6**

Status

Active

Inactive

Active for the period:

Starting: Date: **Oct 23, 2024** Time: **9:03:17 AM**

Ending: Date: **Oct 23, 2024** Time: **9:03:17 AM**

Every day between: from: **9:03:17 AM** to: **9:03:17 AM**

Except these days of the month

Except these days of the week: Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Execution

On startup Time: **11:45:00 PM**

Simple

Hourly

Daily

Weekly

Monthly (day of the month)

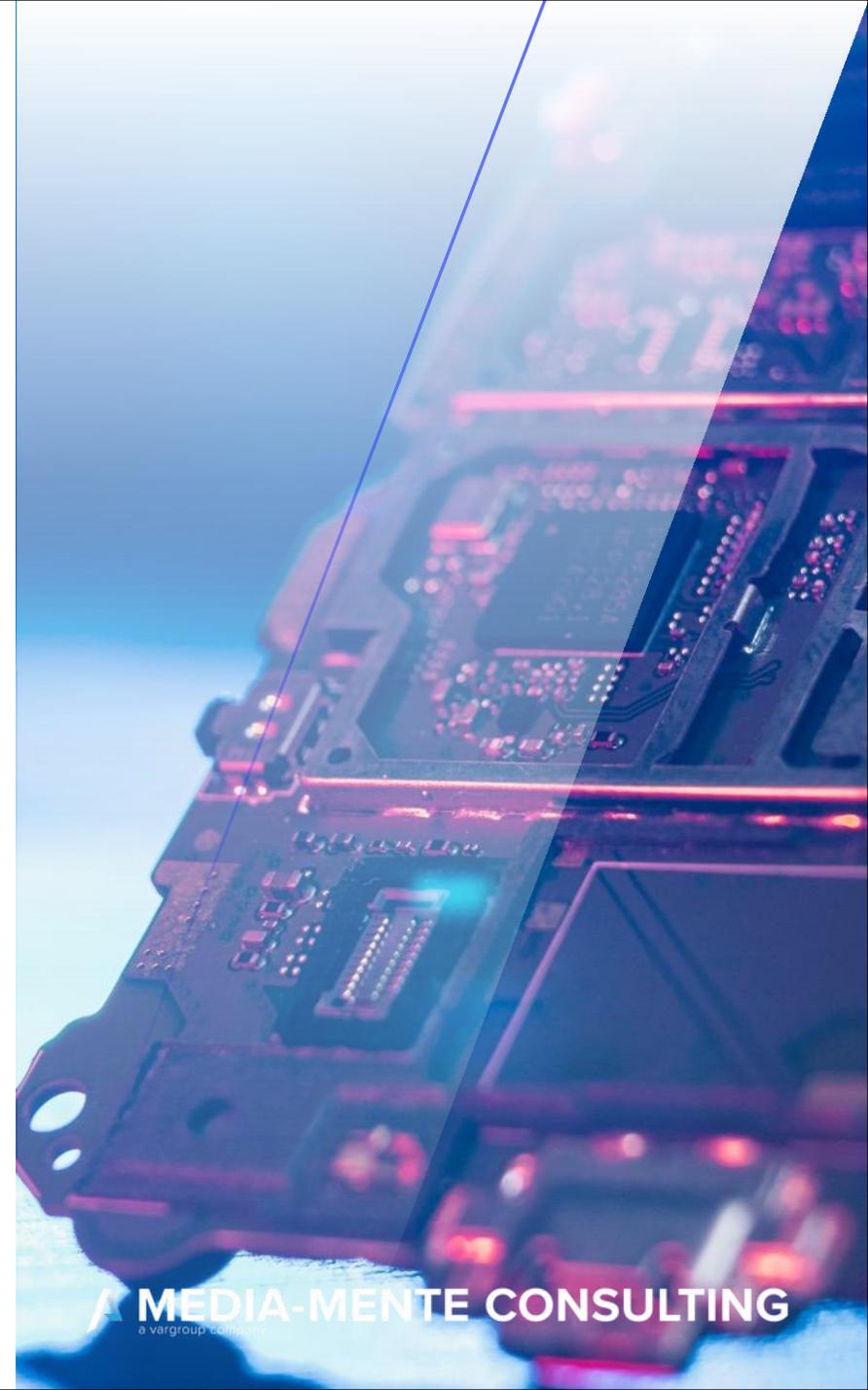
Monthly (week day)

Yearly

Load Plan Scheduling GLOBAL / odi_agent

Setting your Load Plan Variables:

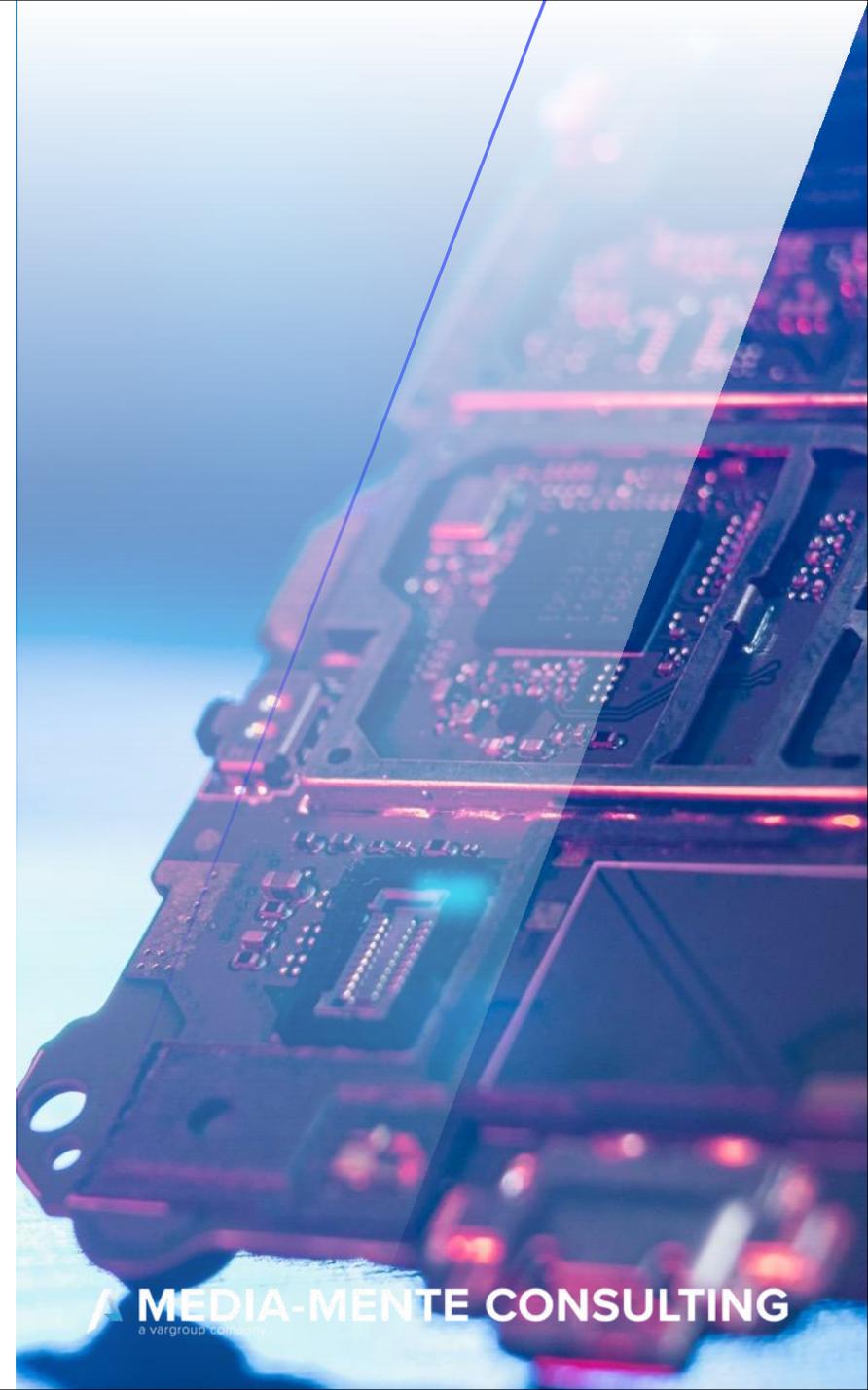
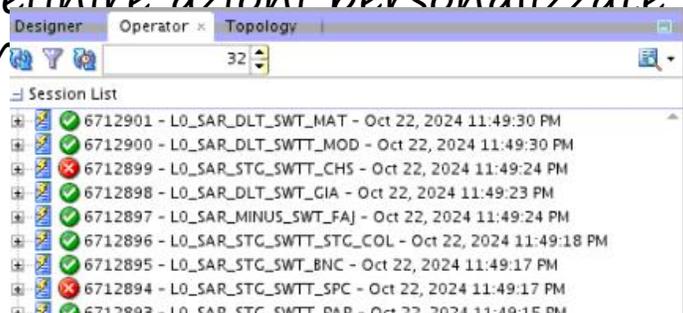
Name	Value
DWH.ANAGRAFICHE	1
DWH.IDENTITY	
DWH.JOBID	
DWH.MOVIMENTI	1
DWH.NULL_DESC	
DWH.NULL_JOBID	
DWH.NULL_NUMBER	
DWH.NULL_VARCHAR	
DWH.NUM_LEVEL	
DWH.PUBLIC_COMPANY_COD	
DWH.RESET_DWH	
DWH.V_RUN_LEVEL_0	1
DWH.V_RUN_LEVEL_1	1
DWH.V_RUN_LEVEL_2	1
DWH.COMPANY_CURRENCY	
DWH.DWH_MODE	
DWH.MAIL_TO	
DWH.MAIL_CC	
DWH.LATENCY	
DWH.JDE	0
DWH.SAR	1
DWH.SELLOUT	1
DWH.ESITO_CHECK_TO_USERS	
DWH.MAIL_USERS_TO	
DWH.MAIL_USERS_CC	
DWH.SELLIN	0
DWH.BUDGET	
DWH.MAIL_USERS_CC_SELLIN	
DWH.BUDGET_SELLOUT	1
DWH.CRP_NAMF	



Monitoraggio

ODI consente di implementare strategie di error handling nei flussi di integrazione utilizzando:

- **ODI Operator:** Per visualizzare lo stato dei job e delle esecuzioni passate.
- **ODI Console:** Interfaccia web che consente di monitorare lo stato dei job, visualizzare i log e le performance.
- Memorizzazione log di esecuzione per ogni job con dettaglio degli errori
- Tracciamento delle performance di caricamento e trasformazione.
- Knowledge Modules con gestione degli errori.
- Tabelle E\$ per registrare i record problematici.
- Possibilità di definire azioni personalizzate in caso di errore, come ripetizioni



Use Cases



Use Case 1: Fashion

Panoramica dell'azienda:
Un'entità leader nel settore della moda con un fatturato > 500 milioni di euro.

Contesto del progetto:
implementazione e manutenzione di un nuovo Enterprise Data Warehouse (EDWH).

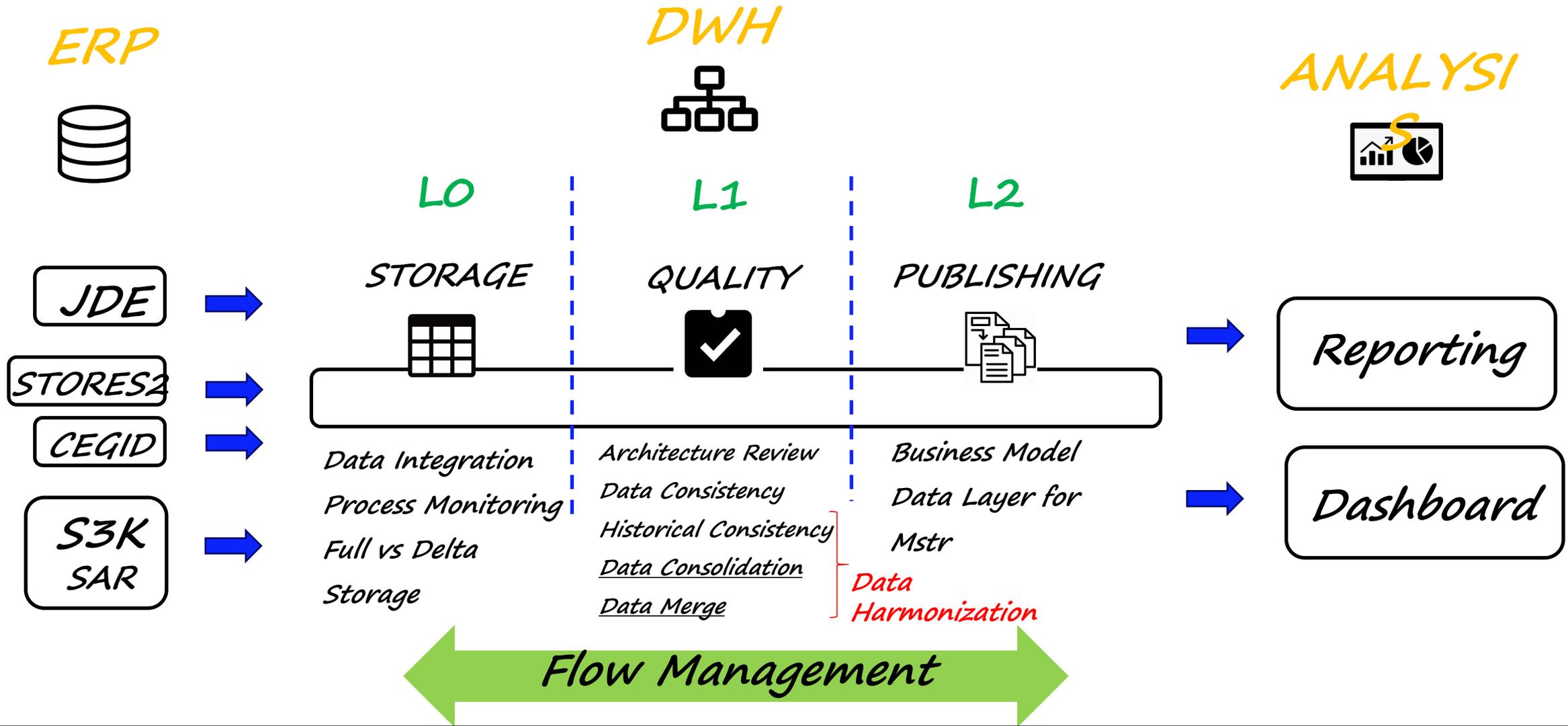
Obiettivo: Generare valore tramite il dato aziendale e supportare funzioni critiche in tutta l'organizzazione.



1. Aumento dell'adozione
2. Integrazione nella roadmap analitica apportando benefici al Planning e Merchandising
3. Implementazione di tecniche avanzate di visualizzazione dei dati per facilitare decisioni
4. Integrazione dati provenienti da sistemi verticali, ERP, PLM, CRM e piattaforme eCommerce
5. Gestione della qualità dei dati e dei master data (MDM) Garantendo elevati standard di qualità e una gestione efficace dei dati
6. Pratiche solide di governance dei dati per mantenere l'integrità e la conformità dei dati
7. Implementate misure rigorose di sicurezza per proteggere le informazioni sensibili



Use Case 1: Fashion



Use Case 2: Pharma

*Panoramica dell'azienda:
Settore farmaceutico con
fatturato > 4 miliardi di
euro.*

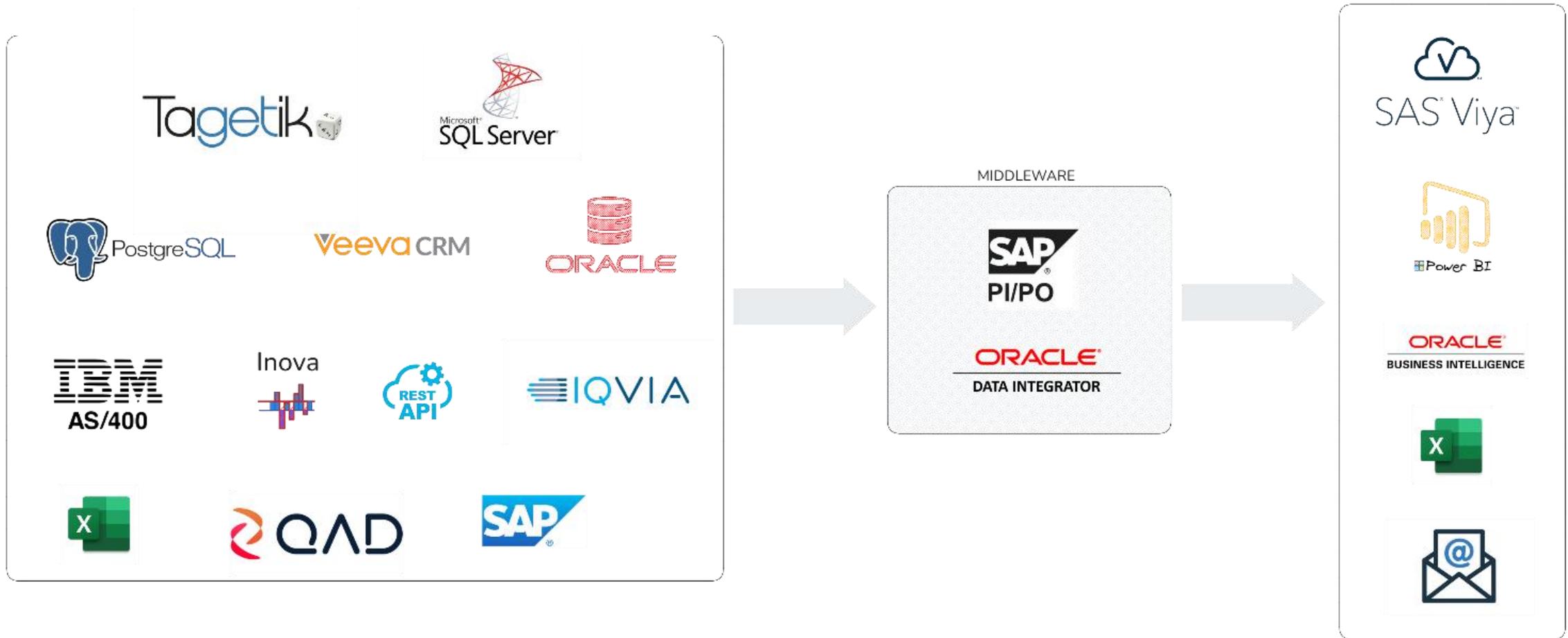
*Contesto del progetto:
Progetto strategico per
stabilire nuove linee guida
per l'integrazione dei dati,
sviluppando con successo
un quadro di governance
dei dati completo,
gestendo meticolosamente
l'orchestrazione dei
processi.*

*Obiettivo: Ridurre
significativamente gli
sforzi di manutenzione ed
elimina gli errori di
integrazione.*



1. Enterprise Data Warehouse (EDWH)
2. Creato un ufficio dedicato alla governance dei dati per supervisionare e applicare politiche e standard.
3. Riduzione degli sforzi di manutenzione del 50%
4. Integrazioni senza errori, garantendo operazioni dati senza interruzioni.
5. Sviluppato e implementato un quadro solido per garantire l'integrità, la conformità dei dati.
6. Manutenzione e aggiornamento regolari dell'infrastruttura dati per supportare gli obiettivi dell'organizzazione.
7. Orchestrazione tra applicazioni per garantire operazioni fluide ed efficienti.
8. Collaborato strettamente con il team di Data Architect per standardizzare le linee guida e le integrazioni in tutta l'organizzazione.

Use Case 2: Pharma



Cloud e Big Data



ODI Cloud e ODI On-Premise

- *ODI On-Premise: Tradizionale installazione di ODI nei server locali, con gestione diretta delle risorse e dell'infrastruttura.*
 - *ODI Cloud: ODI è disponibile anche come servizio cloud all'interno di Oracle Cloud Infrastructure (OCI).*
- *Vantaggi di ODI Cloud:*
- *Scalabilità automatica e flessibilità per gestire picchi di carico.*
 - *Nessuna necessità di gestione diretta dell'infrastruttura.*
 - *Integrazione nativa con altri servizi cloud di Oracle.*



ODI e Big Data

- *Integrazione con tecnologie Big Data:*

ODI offre connettori e Knowledge Modules specifici per gestire i Big Data, tra cui:

- *Hadoop, Spark, HDFS, Hive.*
- *NoSQL (Cassandra, MongoDB).*

- *Funzionalità Big Data di ODI:*

- *Supporto nativo per l'esecuzione di job su cluster Hadoop.*
- *Esecuzione di trasformazioni dati su Spark per alte performance.*
- *Integrazione con ecosistemi di Big Data per gestire grandi volumi di dati.*



Conclusioni



Perché scegliere ODI?

➤ Performance ed Efficienza

Architettura **E-LT** ottimizza i tempi di esecuzione riducendo il carico sui server di origine, le risorse e il supporto cloud-native riducendo i costi operativi a lungo termine rispetto ai competitors.

➤ Integrazione nativa con Oracle

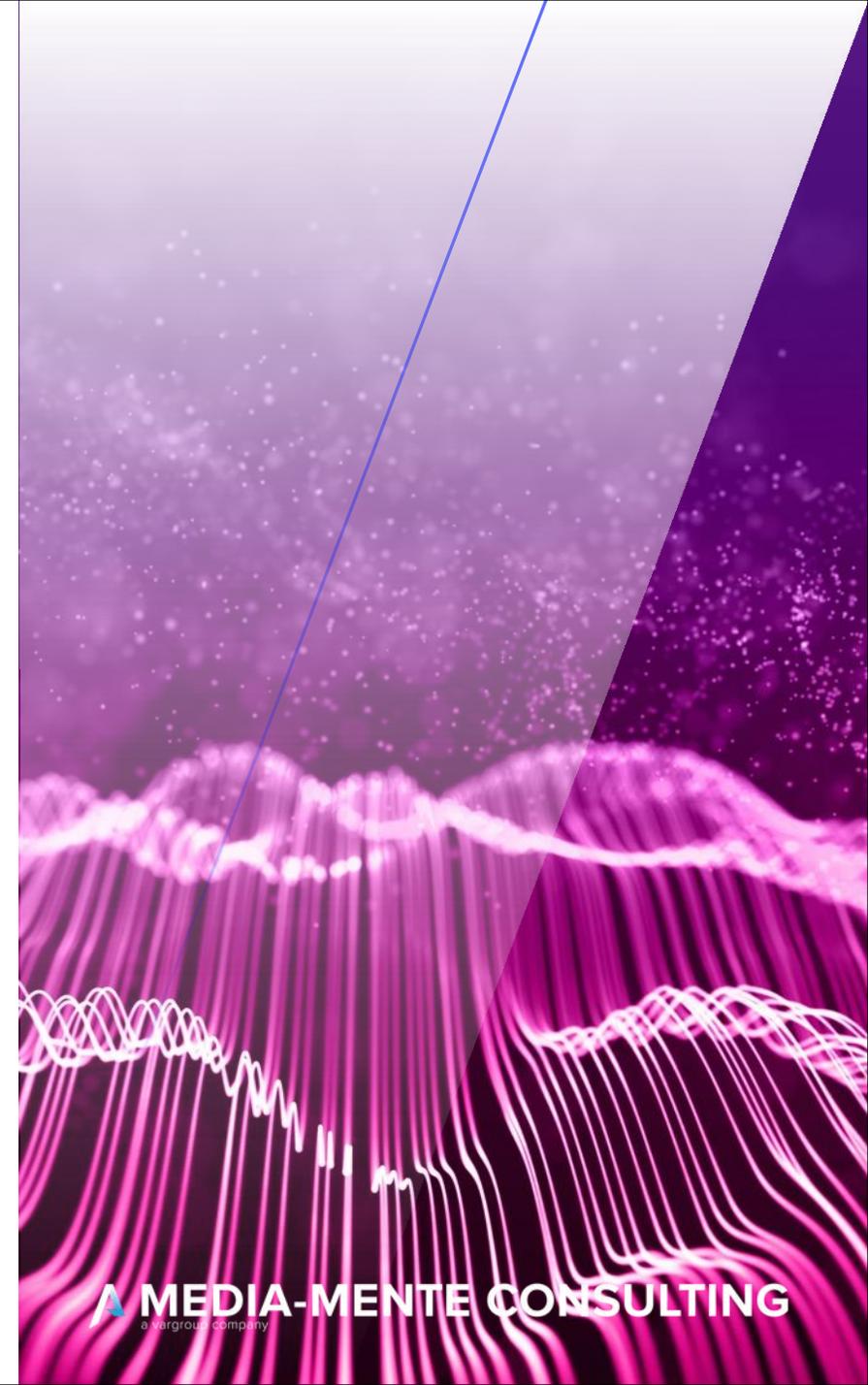
Integrazione perfetta con l'ecosistema Oracle e supporto per database eterogenei, riducendo costi e complessità rispetto a soluzioni non Oracle.

➤ Flessibilità e Scalabilità

ODI permette di configurare processi di integrazione complessi in modo visivo, mantenendo un alto livello di controllo. È progettato per scalare con l'aumento dei dati aziendali.

➤ Sicurezza e Governance Dati

Funzionalità avanzate di sicurezza e governance garantiscono la conformità alle normative più stringenti, offrendo una gestione sicura e tracciabile dei dati.



What's Next?

➤ **Generative AI**

Progettazione e standardizzazione per l'integrazione e le funzionalità di ODI in un ambiente controllato con l'utilizzo di Intelligenza Artificiale

➤ **Scalabilità Avanzata**

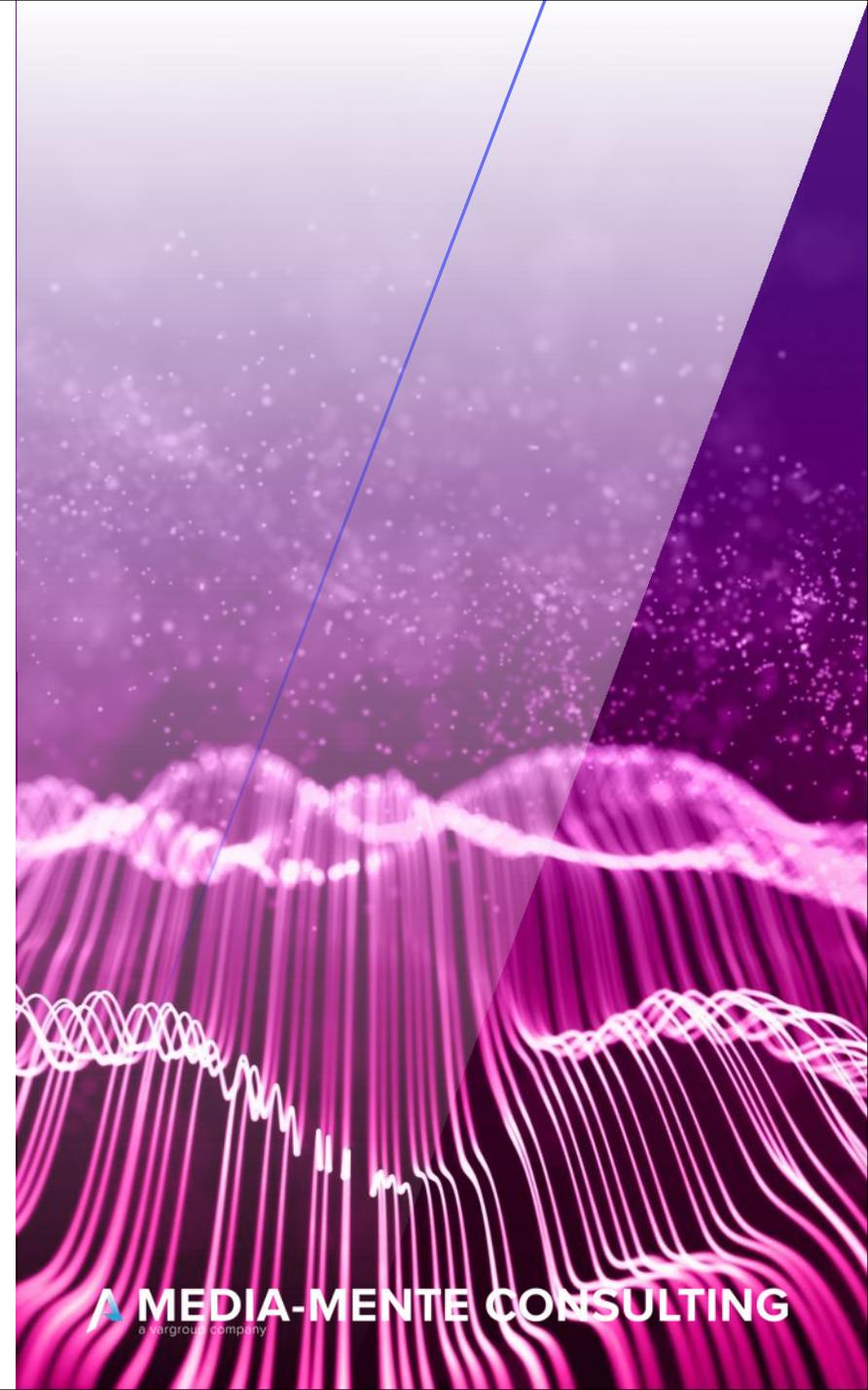
Identificare i processi chiave da automatizzare e pianificare la scalabilità delle operazioni con l'aumento del volume di dati.

➤ **Monitoraggio e Validazione**

Implementazione di strumenti di monitoring e logging avanzato per tracciare performance e feedback dei processi di integrazione con suggerimento in automatico eventuali miglioramenti.

➤ **Integrazione con Infrastruttura Esistente**

Integrazione di ODI con le piattaforme esistenti (ad es. Oracle, SQL Server, Hadoop) utilizzando connettori e configurazioni di data mappings ottimizzati per compatibilità e latenza



Q&

A

24/10/2024

Bregata Luca

l.bregata@mediamenteconsulting.it

