

Event-Driven Data Integration to Automate Elevator Service Contract Booking Between MS Dynamics and Oracle JDE with Operational MDM for Master Data Integration

Shreesha Hegde Kukkuhalli

hegde.shreesha@gmail.com

I. Abstract

With increasing digitization in industries, data driven integration between enterprise systems is critical for automating business processes and improving efficiency. This paper presents a novel approach to automate elevator service contract bookings by integrating Microsoft Dynamics (MS Dynamics) with Oracle JD Edwards EnterpriseOne (JDE) through event-driven data integration. A key aspect of this integration is the utilization of an Operational Master Data Management (MDM) system to ensure consistency of master data across both platforms. This event-driven architecture significantly reduces manual intervention and improves data accuracy by ensuring near real-time synchronization of data across systems.

Paper will also explore a case study where this event driven automation was implemented for a large elevator and escalator manufacturing and services company to reduce overall contract booking turnaround time from 45 days to 5 days and save more than \$1M in labor cost over 3 years.

Keywords

Event-driven integration, master data management (MDM), enterprise resource planning (ERP), MS Dynamics, Oracle JDE, automation, service contract booking, data synchronization.

II. Introduction

The automation of business processes is critical to the efficiency and competitiveness of organizations, particularly when multiple enterprise systems are involved. Many organizations struggle with the manual reconciliation of data between disparate systems, leading to inefficiencies, errors, and delays. This is particularly true in industries where service contracts are central to operations, such as the elevator maintenance industry, which requires precise coordination between customer relationship management (CRM) and enterprise resource planning (ERP) systems.

In this paper, I propose an event-driven integration framework that automates the booking of elevator service contracts between MS Dynamics, which serves as the CRM system, and Oracle JDE, the ERP system responsible for managing contracts and financial operations. To ensure consistency and accuracy, we incorporate a Master Data Management (MDM) layer that synchronizes key master data entities between both systems. The framework employs near real-time event-driven messaging to keep the systems synchronized and provide seamless data flow.

III. Main Body

Background and Related Work

A. Data Integration Challenges in CRM-ERP Systems

Data integration between CRM and ERP systems poses several challenges, including differences in data models, the need for real-time synchronization, and the management of duplicate or inconsistent data. Previous studies have highlighted the complexities involved in integrating systems such as SAP, Oracle, and MS Dynamics due to variations in data standards, APIs, and transaction processing mechanisms.

B. Event-Driven Architectures

Event-driven architectures (EDAs) offer a flexible and scalable way to integrate systems by reacting to real-time changes (events) rather than relying on periodic batch processing. EDAs have proven effective in environments requiring low-latency data propagation, such as stock trading platforms, logistics networks, and financial systems. In the context of this study, an EDA is employed to propagate service contract creation events from MS Dynamics to Oracle JDE, enabling the automated booking of contracts without manual intervention.

C. Master Data Management (MDM)

MDM provides a comprehensive approach to managing master data entities (such as customer, unit (elevator), and contract data) across multiple systems. MDM ensures that each system has access to consistent, validated, and up-to-date data, which is essential in integration scenarios where multiple systems interact with the same entities. For the elevator service industry, maintaining a unified view of customer and contract data is essential to operational efficiency.

System Architecture

The proposed system architecture integrates MS Dynamics with Oracle JDE using an event-driven approach facilitated by an MDM layer.

A. Components of the Architecture

1. **MS Dynamics**

MS Dynamics serves as the front-end CRM system where customer data, service requests, and contracts are created and managed. This system initiates the booking process for elevator service contracts.

2. **Oracle JD Edwards EnterpriseOne**

Oracle JDE functions as the back-end ERP system responsible for service fulfillment, order management, and financial accounting. The service contract information from MS Dynamics is automatically propagated to Oracle JDE via event-driven integration.

3. **Boomi Master Data Hub (Operational Master Data Management Hub)**

The MDM hub acts as the source of truth for master data entities, including customer, product, and contract data. The MDM ensures that both MS Dynamics and Oracle JDE operate on consistent and validated master data.

4. **Boomi Integration Service:** Boomi integration platform as a service acts as low code UI based integration tool to integrate data between all the systems involved. Boomi has built in connectors to connect to MS dynamics, Oracle JDE, Master data hub and Azure service bus.

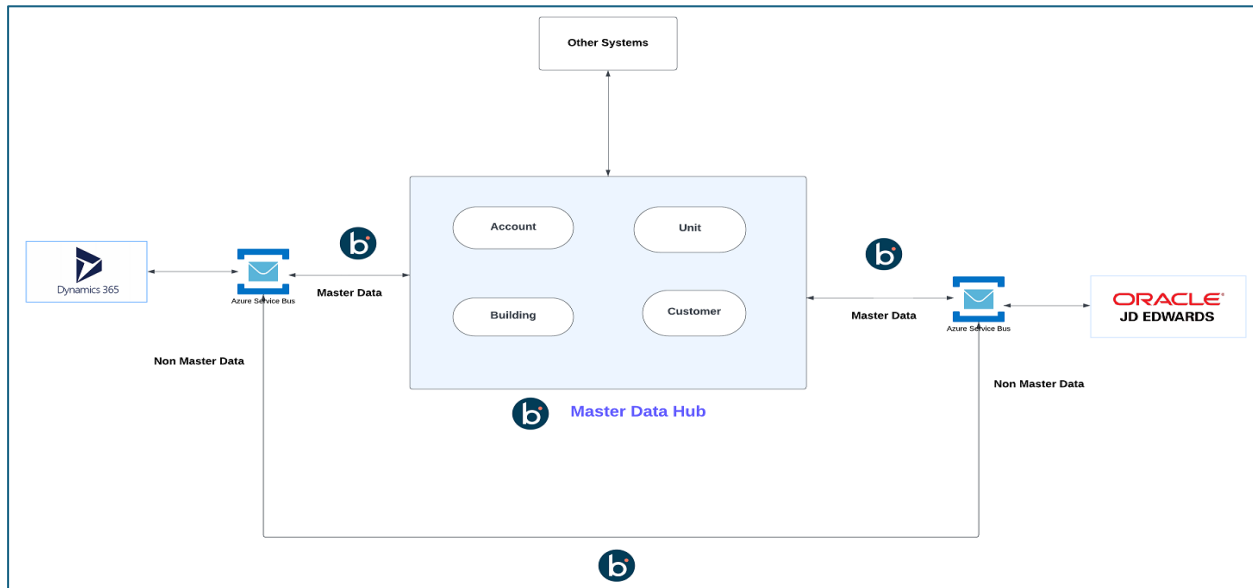
5. Azure Service Bus (Message Bus)

Azure service bus is responsible for storing events generated by MS Dynamics, Oracle JDE and Boomi Master Data Hub. It utilizes event topics for service contract creation, updates, renegotiation, and cancellations.

Sample proposed message format to push data from MS Dynamics to Azure Service Bus:

"MessageType": "MDM Account",	String	
"Country": "US", -- NAA, FR, CH	String	
"CompanyCode": "string", -- NAA, FR, CH etc	String	
"TransactionId": "", --JDE Inegration Job Id	String	
"Account": {}	Array	
"AccountGroup_DataValue": 0,	Integer	Path: customerid_account.gsscore_AccountGroups.gsscore_value
"AccountGroupAutoID": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_autoid
"AccountGroupCode": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_accountgroupcode
"AccountGroup": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_name
"AccountSegment": "string",	String	Path: customerid_account.gsscore_segment
"AccountName": "string",	String	Path: customerid_account.name
"AccountSector": 0,	Integer	Path: customerid_account.gsscore_AccountSector.gsscore_value
"AccountType": 671710000,	Integer	Path: customerid_account.gsscore_accounttype
"AdditionalAddress": "string",	String	Path: customerid_account.gsscore_additionaladdress
"Address1_Fax": "string",	String	Path: customerid_account.address1_fax
"BillingName": "string",	String	Path: customerid_account.gsscore_billingname
"BillingAddressCity": "string",	String	Path: customerid_account.address2_city
"BillingAddressCountry": "string",	String	Path: customerid_account.gsscore_BillingAddressCountry.gsscore_erpcountrycode
"BillingAddressLine1": "string",	String	Path: customerid_account.address2_line1
"BillingAddressLine2": "string",	String	Path: customerid_account.address2_line2
"BillingAddressLine3": "string",	String	Path: customerid_account.address2_line3
"BillingAddressState": "string",	String	Path: customerid_account.gsscore_BillingStateProvince.gsscore_name
"BillingAddressStateCode": "string",	String	Path: customerid_account.gsscore_BillingStateProvince.gsscore_statecode
"BillingAddressZipCode": "string",	String	Path: customerid_account.address2_postalcode
"Branch_Code": "string",	String	Path: customerid_account.gsscore_BranchName.gsscore_servicebranchnumber
"KeyAccountManager": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_KeyAccountManager.internalemailaddress
"KeyAccountManager_FirstName": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_KeyAccountManager.firstname
"KeyAccountManager_LastName": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_KeyAccountManager.lastname
"KeyAccountManager_Employeeid": "string",	String	Path: customerid_account.gsscore_AccountGroups.gsscore_KeyAccountManager.employeeid
"ComplementaryAccount_Name": "string",	String	Path: customerid_account.gsscore_preferredname
"ContactEmailAddress": "string",	String	Path: customerid_account.primarycontactid.emailaddress1
"ContactFirstName": "string",	String	Path: customerid_account.primarycontactid.firstname
"ContactLastName": "string",	String	Path: customerid_account.primarycontactid.lastname
"ContactOwner": "string",	String	Path: customerid_account.primarycontactid.owninguser.internalemailaddress
"ContactOwner_FirstName": "string",	String	Path: customerid_account.primarycontactid.owninguser.firstname
"ContactOwner_LastName": "string",	String	Path: customerid_account.primarycontactid.owninguser.lastname

Architecture diagram



Integration Workflow

The integration workflow follows an event-driven approach where specific triggers in MS Dynamics initiate events that are consumed by Oracle JDE. The following steps outline the workflow for booking an elevator service contract:

1. Service Contract Creation in MS Dynamics

When a new elevator service contract is created in MS Dynamics, an event is published to the Azure service bus. This event includes details such as the building, unit, customer, contract terms, service level agreements (SLAs), and pricing.

2. Event Propagation

Master data generated through events propagate to Oracle JDE through Boomi Master data hub, non-master data generated through events propagate to Oracle JDE through direct integration using Boomi integration service.

3. Master Data Validation by MDM

The MDM hub validates the building, unit customer and contact data against its master records. If the master data is new then new golden records are created in master data hub, if it is existing then golden record is updated. If data provided does not pass the rules defined in master data hub then records are quarantined.

4. Contract Booking in Oracle JDE

Once validated, the master data is consumed by Oracle JDE. The system automatically generates a contract ERP number and initiates the financial processes related to the contract. Notifications are sent back to MS Dynamics upon successful booking.

Case Study

Background: North America operating unit of large global elevator manufacturing and services company was facing severe challenges in service contract booking due to disconnected manual processes pushing the turnaround time to book a contract to 45+ days along with data quality issues resulting in lost revenue opportunity, higher cost and lower customer satisfaction. Company took a step to modernize service contract booking process through end to end automation.

Implementation: Company chose to automate the contract booking process with the architecture described above.

1. Number of interfaces and integration defined between MS Dynamics, Boomi Master data hub and Oracle JDE.
2. Master and non master data sets were identified, inbound and outbound message format was established between all the systems.
3. Integrations are developed and tested for both functional and non-functional requirements such as performance, scalability etc.

Results: End to end automation of service contract booking significantly improved key metrics:

1. 90% reduction in overall service contract booking turnaround time (from 45 days to 5 days).
2. More than 50% decrease in data quality issues related to master and non-master data sets.
3. More than \$1M+ in cost savings over 3 years due to elimination of manual effort.

IV. Conclusion

The event-driven data integration architecture presented in this paper demonstrates a scalable and efficient method for automating service contract bookings between MS Dynamics and Oracle JDE. By incorporating operational MDM layer, the system ensures business critical master data consistency and accuracy across multiple platforms. Future work could explore extending this framework to other business processes and systems, further enhancing the operational efficiency and streamlining processes specifically for large global enterprises operating across multiple regions in the world with disparate tech stack.

References

- M. Smith et al., "Integrating CRM and ERP Systems in Enterprises: Challenges and Solutions," *Journal of Enterprise Systems Integration*, vol. 25, no. 4, pp. 67-78, 2022.
- L. Zhou and K. Bhargava, "Event-Driven Architectures: A Survey," *IEEE Transactions on Software Engineering*, vol. 39, no. 12, pp. 89-102, 2021.
- S. Williams, "Master Data Management in Complex Enterprise Systems," *MDM Journal*, vol. 18, no. 2, pp. 23-34, 2020.
- J. K. Munoz and M. Whitaker, "Master Data Management in the Era of Big Data: Challenges and Solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 590-600, 2022.
- M. Behrens and C. Garcia, "Integrating ERP and CRM Systems Using an Event-Driven Architecture," *Journal of Software Integration and Process Management*, vol. 27, no. 4, pp. 150-165, 2021.
- S. Ghosh, "Automating Business Processes in ERP Systems Through Event-Driven Architecture: A Study of Industry Implementations," *Enterprise Information Systems Journal*, vol. 29, no. 5, pp. 103-112, 2021.