RESEARCH ARTICLE OPEN ACCESS

Reliable Dynamic Data Integration Approach for SAAS Application with Their On-Premises Systems

K. Abdur Rahman Khan [1], Dr. P. Suryanarayana Babu [2]
Lecturer in Computer Science [1], Research Scholar
Department of Computer Science, Osmania College
Principal [2], Research Supervisor
Department of Computer Science, Sri Vijayadurga Degree College
Rayalaseema University
Kurnool-518002. Andhra Pradesh. India

ABSTRACT

Efficient and effective integration of SAAS applications with on premises enterprise systems, such as Enterprise Resource planning (ERP), Custom applications, and CRM (Customer relation management) applications is always a critical and challenging task to provide customer satisfaction. In this paper, a study is made to summarize the data synchronization and integration with on-premises enterprise applications from various aspects. Based on the other various patterns a proposal is made towards a novel real-time integration approach to perform critical and complex data synchronization easily. The seamless data integration helps the organization to reach high levels of operational consistency, reliability and reusability.

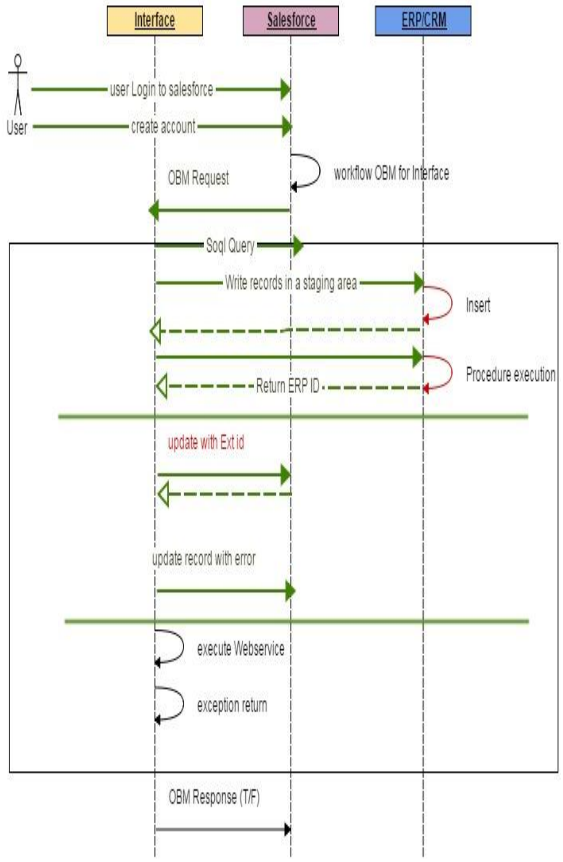
***Keywords*:-** CRM, ERP, SAAS

I. INTRODUCTION

SAAS (software as a service), a customer contracts to use any cloud based application, such as Enterprise Resource Planning (ERP) or Customer Relation Management (CRM), hosted by a third party, rather than purchase a software original license and installing the application software on its own environment. Just as consumers can check e-mail or use mapping programs/ applications with their related Web browsers, so that customers can access their enterprise business applications over the web/Internet [1]. A novel enterprise uses various many applications across their premises which are not default designated with out-of-the-box feature. Each and every application has various layers namely data, presentation, business logic, and security layers. Security layer integration is one of the key aspects for making application integration across the enterprise. Cloud security can be achieved with various development tools (security source code scanner, chimera).we analyzed various data integration patterns [2]. Real time Data layer integration provides the usage of common shared data and effectively manages data across the multiple applications. Data layers integration can achieve with various methodologies (like bulk data transfer API's, Language specific tool kits (ex: java, java script, Ruby), Cloud-to-Cloud (Google, Facebook), and open API's (SOAP, REST)). Business Logic often spans across the various enterprise application to implement the complete end-to-end business process functionality we have to enhance the related existed business logic from one system to another. Business logic integration can be achieved with messaging systems, web services, and specific language callouts; Presentation layer integration which helps the organization to access various types of templates, canvas, connect with social media, accessing existed web pages across the multiple applications. Presentation layer integration can be achieved with various software development kits (SDK's) [3].

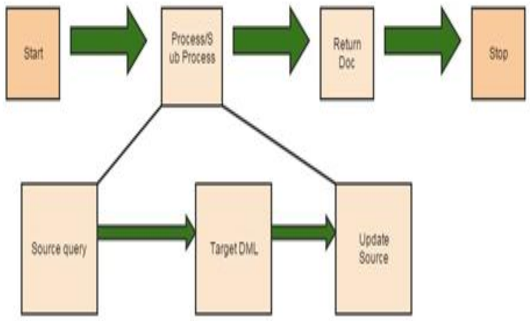
II. PROPOSED SOLUTION

As a SAAS User, I want to synchronize the newly created particular object record from SFDC with any other legacy application so that I can avoid manual re-creation of the same record information in the destination application.in this proposed system we used source system as Salesforce CRM, Destination system as Siebel, Source system object as Account .Middleware Used as Cast Iron and integration pattern as Outbound messaging.



When the user creates an Account in SFDC, Workflow will verify the trigger condition. If the condition is met, it will fire OBM to Cast Iron. If the response from Cast Iron is *true*, SFDC will stop retrying the transaction and remove the message from Retry Queue. If the OBM is not received and processed by CI, SFDC will retry progressively for 24 hours, after several unsuccessful attempts, SFDC willmove the message to Retry Queue for 7 days. Failures to process once the OBM has been received by CI will be handled via the integration handler and either recording into the Oracle Error tables and/or email notification to appropriatecontacts.

For each Account record that is successfully created, submitthe ERP_ID to the Cast-ironWeb Service for CreatingAddresses/Sites within Oracle. The Address/Site Creation Webservice will perform all processing internally, writing the Site external ID back to SFDC and will not require the Accountservice to perform any other action on behalf of the AddressWS. It will however return a success/failure message to the calling orchestration. The account orchestration will store those errors for administrative review (store in variable, but donot return to SFDC). A failure of the Address web service to process one or more addresses will not result in a failure message (via OBMresponse) being sent back to SFDC to occur, unless the orchestration has a fatal error and the Address orchestration was not able to process any records. Fatal Errors should resultin an failure status being returned to the Integration Error Handler. All other errors will be managed and processed internally to the Address Orchestration.The cast Iron Interface framework is illustrated in the following diagram.



Create STAGING table for Cast Iron when Account is created in SFDC Create SiebelStored Procedure to process any staging table entries and load them into main tables. On successful insert, delete the corresponding row in STAGING table. ProvideStored Procedure name, input and output parameters to Developer Company Create view of Main tables to allow data to be viewed by CIAdd a field to Siebelto hold the SFDC ID for the account record.

III. CONCLUSION

These patterns address the requirement to synchronize data that resides in two or more systems so that both systems always contain timely and meaningful data. Data integration is often the simplest type of integration to implement, but requires proper information management techniques to make the solution sustainable and cost-effective. Such techniques often include aspects of Master Data Management (MDM), data governance, mastering, de-duplication, data flow design, and others.

REFERENCES

[1] A Dubey and D Wagle , "Delivering software as a service" , *The McKinsey Quarterly, Web exclusive* , 2007

[2] https://www.mulesoft.com/lp/whitepaper/saas/salesforce-integration-patterns

[3] https://developer.salesforce.com/docs/atlas.enus.integration_patterns_and_practices.meta/integration_patterns_and_practices/

[4] F Zhu , "Dynamic data integration using Web services" , *Proceedings IEEE International Conference on Web Services* , pp.262 -269

[5] J Ullberg, D Chen and P Johnson *Enterprise Interoperability* , pp.13 -24, 2009 : Springer Berlin Heidelberg

[6] P Ziegler and K R Dittrich , *Conceptual Modelling in Information Systems Engineering* , pp.39 -58 , 2007 : Springer Berlin Heidelberg

[7] A Halevy, A Rajaraman and J Ordille, "Data integration: the teenage years", *VLDB Endowment*, pp.9 -16

[8] P A Bernstein and L M Haas, "Information integration in the enterprise", *Communications of the ACM* , vol. 51 , pp.72 -79 , 2008

[9] A Schwinn and J Schelp , *Business Information Systems* , pp.232 -238, 2003 :Colorado Springs

[10] G Hohpe and B Woolf , "Enterprise integration patterns", *9th Conference on Pattern Language of Programs* , pp.1 -9

[11] E Thoo, T Friedman and M A Beyer , "Magic Quadrant for Data Integration Tools" , *Gartner* , 2012

[12] T Scheibler, R Mietzner and F Leymann, "EAI as a Service-Combining the Power of Executable EAI Patterns and SaaS", *Enterprise Distributed Object Computing Conference* , vol. 1 , pp.07 -116

[13] N V S Kolluru and N Mantha , "Cloud Integration - Strategy to Connect Applications to Cloud" , *Annual IEEE India Conference (INDICON).*