SaaS Integration in Supply Chain Management Companies

[1]Adarsh S Jamadagni, [2]Dr. Kariyappa B. S

[1]Student, [2]Professor
R. V. College of Engineering

***Abstract*: In Supply Chain Management based software companies, data is useful in two types of applications. Some applications are useful for clients in prediction of entities such as demand for products and transportation. Other applications help the client in the real time fulfilment of orders, managing workforce, managing warehouses and transportation, etc. Without historical data, predicting or forecasting is merely impossible and without current data, the management of real time events is not possible. In such companies, clients often want more than one type of software solution. Thus integrating all the different software solutions sold to the client into one software bundle is key in delivering the products to the customer. In this review, we shall see how data is accepted from the client, the different processes done to the data so it is finally ready to be utilised by the supply chain solution software.**

***Keywords*: SaaS, Software, Integration, Supply-chain**

1. Introduction

Supply Chain Management based software companies build software that help the clients in optimising the flow of goods from them being raw materials to products that are finished and ready to be taken delivery of. Instead of building an end to end supply chain management package, the companies often build software packages that offer solutions to one particular problem in the supply chain. As a result, client companies can pick and choose which software solution they want based on their requirements. If a client chooses multiple software solutions, the company must then integrate all chosen software solutions into one software bundle.
The above task of integration is tedious as each client might have their own requirements in software solutions and each client will have different types of data and different formats of data and all of this data must ultimately be processed to make useful information which acts as an input to the software solution that either help in forecasting or in fulfilment of orders.

2. Methodology

In such companies, the task of software integration is achieved by the deployment of certain specific ’Services’. These Services are applications that reside on the clouds and each Service has a certain specific tasks. The flow of data through these Services is indicated in the block diagram below followed by a brief summary of what tasks each of these Services execute.

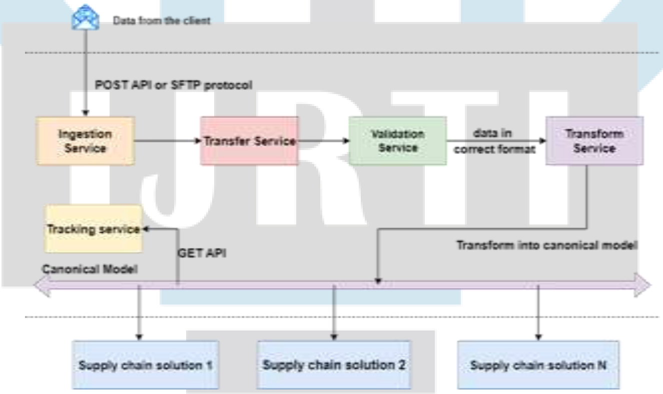


Figure 1. Flow of data

1. Ingestion Service: Ingestion Service enables clients/customers to share data with the cloud. Clients can share data to the cloud by dropping the files at the SFTP location or by invoking the POST API. The Ingestion Service also splits huge files into multiple smaller files and it also notifies the Transfer Service to pick up the files through a Kakfa notification.
2. Transfer Service: Reads files from file storage based on Kafka notification. It parses CSV data into corresponding tables in the ’Staging’ tables. It also notifies the Validation Service that the data is ready for validation.
3. Validation Service: Validation Service reads data from the Staging tables and runs validation checks. These validation checks consist of some standard checks and some custom checks that vary from project to project. It generates log files and reports with errors for the client to rectify their error. It also sends a notification to the Transform s stating that the data is ready for transformation.
4. Transform Service: Transform Service picks up the data from the Curated database and transforms the data into canonical messages (A canonical message is a message in a particular message format that all the supply chain solution packages understand.). Then this transformed message is sent to the supply chain solution. The transformed data is stored in the ’Curated’ tables.
5. Tracking Service: By invoking the GET API from the tracking Service, client can observe the status of the transformed message i.e, if it accepted by the supply chain solution or not.

Staging and Curated tables

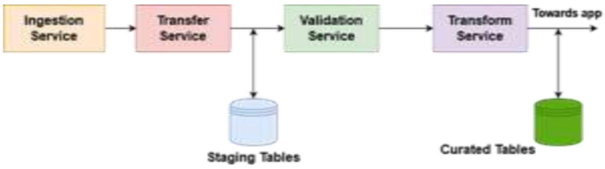
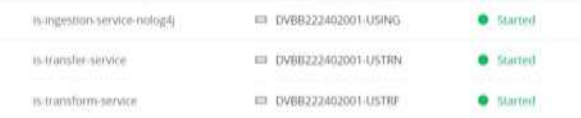


Figure 2. Location of Staging and Curated tables

Staging tables are a set of tables that appear before the Validation Service and after the Transfer Service. A copy of data is stored in this location. This is useful in the case of data not being validated by the Validation Service as one can always look at the data in the Staging tables and logs of the Validation Service and change what is required to get the data validated.
Data is filled in the Curated tables after the Transform Service. In the case that the Supply Chain Solution rejects the data, it is always easy to view the Curated tables and to rectify the errors.

3. Result Discussion

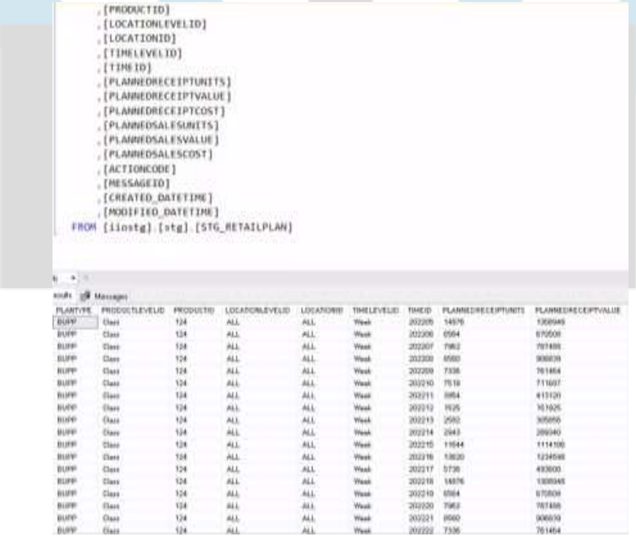
The resulting software integration is not only faster to develop, it also needs lesser manpower. If the clients decide to buy more software solutions down the road, then it would be easier to just make configuration changes to allow for one or more software solutions rather than building the entire software integration from scratch.
The below figure indicates the Ingestion, Transfer and Transform Service ”started” on MuleSoft’s Anypoint Platform. These Services can only be deployed if they are error free. The fact that it is running on a Mule server indicates that these Services are successfully deployed.



is-ingestion-service-nolog4j	DVBB222402001-USING	● Started
is-transfer-service	DVBB222402001-USTRN	● Started
is-transform-service	DVBB222402001-USTRF	● Started

Figure 3. Ingestion Service, Transfer Service and Transform Service ”started” on MuleSoft’s Anypoint Platform

The below figure indicates data being populated in Staging tables. This can only happen if the Ingestion Service and Transfer Service are functioning properly and as a result, data has reached the Staging tables. It is observed that the command has ”[STG]”. This indicates that this is infact, a Staging table.

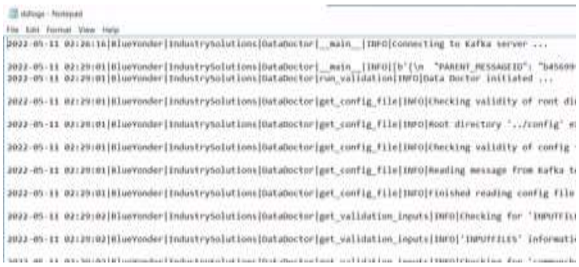


```
      ,[PRODUCTID]
      ,[LOCATIONLEVELID]
      ,[LOCATIONID]
      ,[TIMELEVELID]
      ,[TIMEID]
      ,[PLANNEDRECEIPTUNITS]
      ,[PLANNEDRECEIPTVALUE]
      ,[PLANNEDRECEIPTCOST]
      ,[PLANNEDSALESUNITS]
      ,[PLANNEDSALESVALUE]
      ,[PLANNEDSALESCOST]
      ,[ACTIONCODE]
      ,[MESSAGEID]
      ,[CREATED_DATETIME]
      ,[MODIFIED_DATETIME]
  FROM [iiostg].[stg].[STG_RETAILPLAN]
```

PLANTYPE	PRODUCTLEVELID	PRODUCTID	LOCATIONLEVELID	LOCATIONID	TIMELEVELID	TIMEID	PLANNEDRECEIPTUNITS	PLANNEDRECEIPTVALUE
BUFF	Class	124	ALL	ALL	Week	202205	14576	1308945
BUFF	Class	124	ALL	ALL	Week	202206	6564	670508
BUFF	Class	124	ALL	ALL	Week	202207	7962	787486
BUFF	Class	124	ALL	ALL	Week	202208	9560	906639
BUFF	Class	124	ALL	ALL	Week	202209	7336	761464
BUFF	Class	124	ALL	ALL	Week	202210	7519	711607
BUFF	Class	124	ALL	ALL	Week	202211	3864	413120
BUFF	Class	124	ALL	ALL	Week	202212	1925	161925
BUFF	Class	124	ALL	ALL	Week	202213	2582	305856
BUFF	Class	124	ALL	ALL	Week	202214	2943	289340
BUFF	Class	124	ALL	ALL	Week	202215	11644	1114100
BUFF	Class	124	ALL	ALL	Week	202216	13620	1234548
BUFF	Class	124	ALL	ALL	Week	202217	5736	493600
BUFF	Class	124	ALL	ALL	Week	202218	14876	1308945
BUFF	Class	124	ALL	ALL	Week	202219	6564	670508
BUFF	Class	124	ALL	ALL	Week	202220	7962	787486
BUFF	Class	124	ALL	ALL	Week	202221	9560	906639
BUFF	Class	124	ALL	ALL	Week	202222	7336	761464

Figure 4. Staging tables

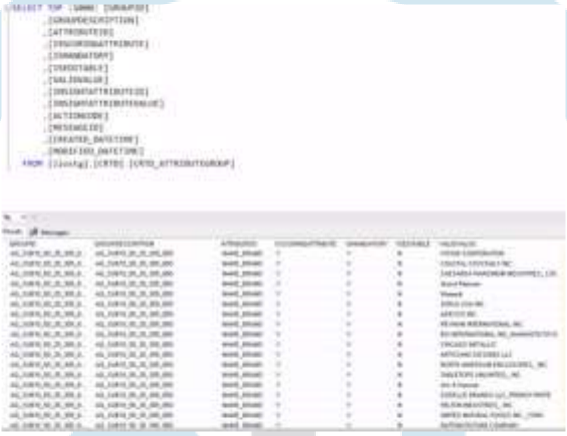
The Validation Service is a bunch of Python programs that check if the data is following standards. This may include Primary Key Constraints, Foreign Key constraints, Missing Value Constraints, Mandatory Value Constraints, etc. After the completion of the checks, certain log files are generated to inform the developers if the data has been validated successfully or not. The below figure is such an example for a log file. This also indicates that the Validation Service is operational.



```
2022-05-11 02:26:16|BlueYonder|IndustrySolutions|DataDoctor|__main__|INFO|Connecting to Kafka server ...

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|__main__|INFO|[b'{\n  "PARENT_MESSAGEID": "b4569
2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|run_validation|INFO|Data Doctor initiated ...

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|get_config_file|INFO|Checking validity of root di

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|get_config_file|INFO|Root directory '../config' e

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|get_config_file|INFO|Checking validity of config 

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|get_config_file|INFO|Reading message from Kafka t

2022-05-11 02:29:01|BlueYonder|IndustrySolutions|DataDoctor|get_config_file|INFO|Finished reading config file

2022-05-11 02:29:02|BlueYonder|IndustrySolutions|DataDoctor|get_validation_inputs|INFO|Checking for 'INPUTFIL

2022-05-11 02:29:02|BlueYonder|IndustrySolutions|DataDoctor|get_validation_inputs|INFO|'INPUTFILES' informati
```

Figure 5. Log file from the Validation Service

The below figure indicates data being populated in Curated tables. This can only happen if the Validation Service and Transform Service are functioning properly and as a result, data has reached the Curated tables. It is observed that the command has ”[CRTD]”. This indicates that this is in fact, a Curated table.



```
SELECT TOP (1000) [GROUPID]
      ,[GROUPDESCRIPTION]
      ,[ATTRIBUTEID]
      ,[illegible]
      ,[ISMANDATORY]
      ,[ISEDITABLE]
      ,[VALIDVALUE]
      ,[illegible]
      ,[illegible]
      ,[ACTIONCODE]
      ,[MESSAGEID]
      ,[CREATED_DATETIME]
      ,[MODIFIED_DATETIME]
  FROM [ilostg].[CRTD].[CRTD_ATTRIBUTEGROUP]
```

It can be concluded that Ingestion Service, Transfer Service and Transform Service are operational from 3. An inference can be drawn from 4 that Ingestion Service and Transfer Service are working as expected as data has reached the Staging tables. It can also be seen that the Validation Service is operating as expected due to 5. Finally, it is observed from 3 that Validation Service and Transform Service are working as expected as the data has reached the Curated tables.

By looking at the above paragraph, we can conclude that the above design for Software Integration is deployed and fully operational.

4. Conclusions

By adopting this methodology in software integration, we arrive at a solution that is modular and is applicable to all supply chain solutions within a company. Based on project and client requirements, configuration changes have to be made in the configuration files of each Service in the case of the client being able to supply recommended certain parameters. Apart from these two cases, the base repositories but not required information or if a client cannot give data on could be used repetitively. Therefore, it would consume less time and would require less workforce to deploy this method for software integration.

References

[1]S. Y M, S. H R, and R. Nagapadma, “Survey Paper: Framework of REST APIs,” vol. 07, no. 06. Mysuru , Karnataka, Jun. 0, 2020.

[2]N. Kumar, “Secure File Transfer Protocol (SFTP) and its Advantages — Encryption Consulting,” Encryption Consulting — Encryption Consulting, Nov. 27, 2021. https://www.encryptionconsulting.com/secure-file-tran sfer-protocol-sftp-and-its-advantages/ (accessed Apr. 22, 2022).

[3]R. Nitnaware, “Basic Fundamental of Python Programming Language and The Bright Future.” 0 0, 2019.

[4]S. Singh and N. Singh, “Containers & Docker: Emerging roles & future of Cloud technology.” 0 0, 2016. doi: 10.1109/ICATCCT.2016.7912109.

[5]A. Tesliuk, S. Bobkov, V. Ilyin, A. Novikov,
A. Poyda, and V. Velikhov, “Kubernetes Container Orchestration as a Framework for Flexible and Effective Scientific Data Analysis.” 0 0, 2019. doi: doi: 10.1109/ISPRAS47671.2019.00016.

[6]K. Uyanık, G. & Güler, and Neşe, “A Study on Multiple Linear Regression Analysis. Procedia - Social and Behavioral Sciences.” 0 0, 2013. doi: 10.1016/j.sbspro.2013.12.027.

[7]B. R. Hiraman, C. V. M, and K. A. C, “A Study of Apache Kafka in Big Data Stream Processing.” 0 0, 2018. doi: 10.1109/ICICET.2018.8533771.

[8]Y. Silva, A. Silva, I. & Queiroz, and Michelle, “SQL: From Traditional Databases to Big Data.” 0 0, 2016. doi: 413-418. 10.1145/2839509.2844560.

[9]B. Raza, A. Mateen, M. Sher, M. M. Awais, and T. Hussain, “Autonomicity in Oracle Database Management System.”

0 0, 2010. doi:10.1109/DSDE.2010.72.
[10]T. Singh, " A Survey on Java Programming Language and Methods of Improvisation." 0 0, 2017. [11]R. Henjes, D. Schlosser, M. Menth, and V.Himmler, "Throughput Performance of the ActiveMQ JMS Server." Springer, Berlin, , 0 0, 2007. doi: 10.1007/978-3-540-69962-0$_1$0.
[12]Building a Digital Platform to Lead in the API
Economy Resource. Mulesoft White Paper.
[13]J. Fremantle, K. Fremantle, P. Rich, and B. Rich, "A history and future of Web APIs." 0 0, 2014. doi: 56. 10.1515/itit-2013-1035.