IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

A Novel information retrieval mechanism with secure authentication for third party applications in cloud based integration

G. Neelima[1*], I. Ramesh Babu[2]

[1]Department of CSE, Acharya Nagarjuna University, Andhra Pradesh, India
[2]Department of CSE, Acharya Nagarjuna University, Andhra Pradesh, India

Available online at: www.ijcseonline.org

Accepted: 19/May/2018, Published: 31/May/2018

Abstract- Cloud computing is a potential and assuring IT technique which helps the users to provide shared computing resources on demand. In recent days the companies are depending up on cloud resources for storage and processing of data. As the control of company's data is handed to the third party the primary factors under utmost concern at the time of information retrieval are security and authentication. Authentication is essential in integration flows between applications for a suite of cloud services enabling users to deploy integrations with security. In my research paper we have used token based OAuth mechanism to provide authentication and query based framework for information retrieval. OAuth is a very secured authentication mechanism that is used to access the data from the source system. Secure retrieval mechanism is used for pulling the data from the cloud. The authentication technique used in our research paper is enhanced by using handshaking process with the source system and token generation in encrypted format. Analysis and implementation of this mechanism in the demo system illustrates its level of security.

Keywords: Cloud Integrations, OAuth Mechanism, OData Framework

I. INTRODUCTION

It is the era of cloud computing which is being used for information processing. It provides remarkable technological opportunities and besides economic benefits. In recent days many organizations and individuals are relying upon cloud platform for data storage and also as their publishing environment. Two types of clouds i.e. public and private clouds are being used by many business applications. While using the cloud services information chaos may result from organization's information management [1]. These kinds of upheavals come from the inconsistency between the internal data and remote data in the public cloud. The management risks are caused by the inconsistency of associated access permission controls in different copies. Organizations are getting into hazards of exposure of their information in the cloud without proper security protection and suffer the risk of unauthorized risk and editing.

Organization's data that is operated from the cloud databases generally confronts the problem of data consistency during the information life cycle. Preserving data integrity is the primary challenge that is encountered by the organizations when they maintain and manage their data from the clouds. The actual data and its representation should be stored in such a manner that the original value can be retrieved without have been read/edited by unauthorized. It is possible by providing access rights, authorization and authentication to the data. Faithfull implementation of security policy and access rights policy is to be done to preserve the security and authentication of cloud data [5]. The primitive operations that are performed by the organization to maintain its data in cloud are backup cloud data, update cloud data and update data permissions. Our research paper focusses on retrieval of organization's data that has been stored in the cloud. With the proposed method high level of authentication is provided and secure data retrieval is done using a middleware integration tool. Authentication is provided by the user to the third party to retrieve data from the cloud with OAuth process and using encrypted key as token. Integration between the cloud and user is established and required data is retrieved as demanded by the queries given by the user.

II. RELATED WORK

Johannes K. Chiang, Eric H.-W. Yen and Yen-Hua Chen has designed a method to solve the authentication and authorization problem by hybrid clouds [1]. They have focused on the implementation of a systematic approach to make the authentication, authorization synchronization and the updates of the access rights of files in local or private cloud.

© 2018, IJCSE All Rights Reserved

Qingqing Xie, Liangmin Wang and Hong Zhong have proposed a novel non-repudiable query answer authentication scheme over anonymous cloud data [7]. They have introduced the non-repudiation protocol into query answer transmitting procedure.

In my research work we have proposed a method which is meant to solve authentication and intruder problem when the user accesses the third-party cloud (SUCCESS FACTORS). We have also designed a framework to retrieve the data from cloud in a very secured manner. Dell Boomi tool is used to integrate the parties involved in cloud data access.

III. MOTIVE AND RESEARCH GOAL

Our research methodology aims at providing security control and authorization when retrieving data from the cloud. OAuth mechanism is used along with secure retrieval mechanism to achieve cloud standardization of authentication and file authorization, our research is to develop cloud data access right management and secure data retrieval in nearly all kind of cloud environment. In our research paper we have concentrated on four primary aspects of cloud data. They are

1. Storing data consistently to the cloud
2. Providing user authentication
3. Secure retrieval of cloud data
4. Back up of data

High level of security is needed while accessing data from the cloud applications. Robust methods are to be used to provide authorization, authentication and security to the data. OAuth mechanism is used before generating queries to the database in cloud. This mechanism includes generation of the token for the user given by the supplied scope. To generate a token the third party needs public key (certificate) and the client application will have the private key. After completing secured authentication mechanism secured connections are established between client and third party that are meant to share data in the cloud [5]. For this integration we have used Dell Boomi, a middle ware tool. Dell Boomi tool is used to securely retrieve the necessary files from the cloud. Dell Boomi is an on-demand integration tool for connecting cloud and on premises applications and data. This tool helps us to transfer the data between cloud and third-party applications with utmost security Common OData Framework to Run Queries. In this paper we have used OAuth for getting the data from the SuccessFactors using queries. OAuth is the type of Authentication process, where we are going to access the data from the source system, before getting the required response, Initially, we need to do some handshaking process with the source system to get the data.

Methodology Used

1. Request data from SuccessFactors
2. Request the source system to generate a token
3. Provide access right to pull the data from the cloud with that token using key as bearer token
4. Pull the data securely from the cloud using Dell Boomi middle ware tool

III. ANALYSIS AND IMPLEMENTATION

The OData API is SuccessFactors Web Services. API based on OData protocol is meant to provide access to data in the SuccessFactors system. The API is data centric. This API provides methods for CRUD access i.e. Create, Read, Update and Delete. This is best used for frequent or real-time requests for small amounts of data. Large data requests are better handled by batch FTP processes. This OData API is used to configure entities. Each module used in this can be accessed using its own set of entities.

© 2018, IJCSE All Rights Reserved

A. Generation of OAuth token

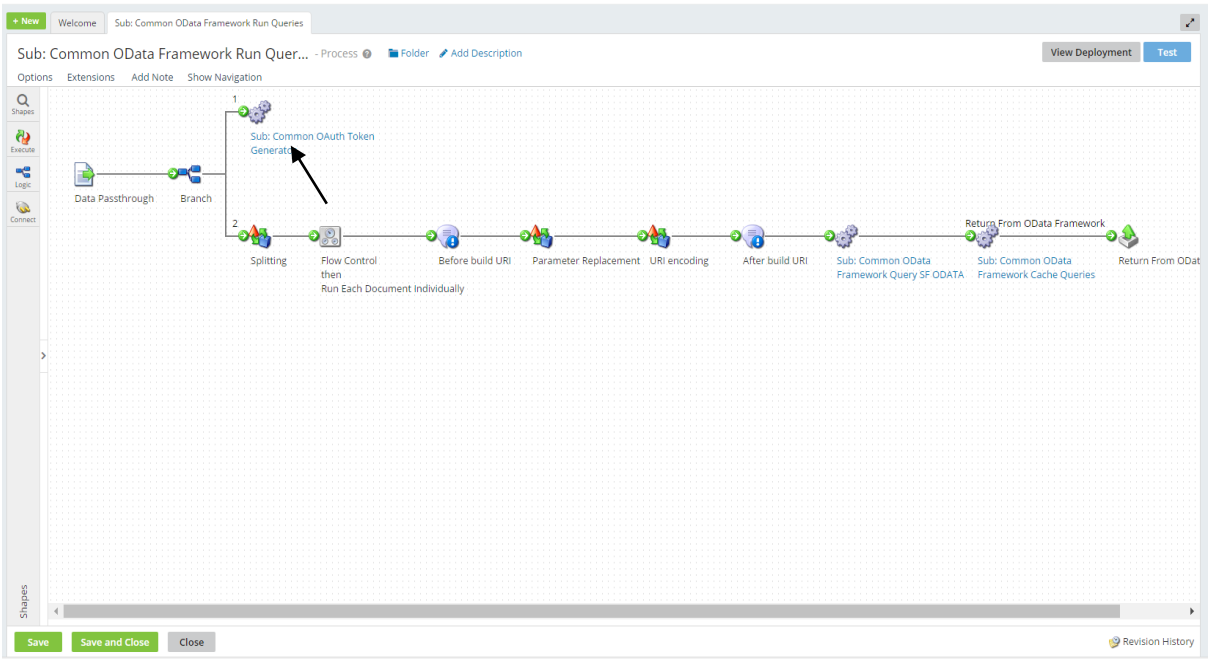


Fig1. Paths to generate OAuth token

Once the Bearer token is created it will expire in 24 hrs., so that we can use the token to get the data for 24 hours from the time of token generation. The above screen shot shows two paths in which first path generates the OAuth token to pull the data from SF, In the token generator Sub process, the token is generated and verifies for Expiry time for every process execution (as shown in the below screen shot), if the token expires, new token will be generated and it is used for next 24 hours.

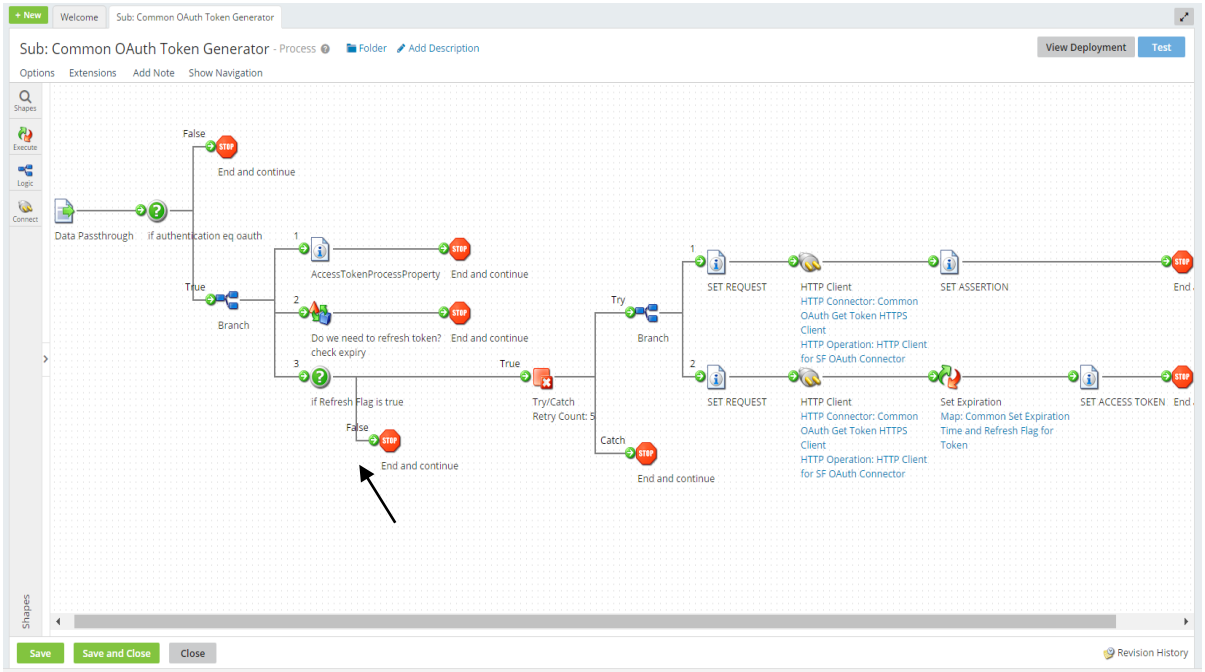


Fig 2. First Path in Token Generation

The second path of "Sub: Common OData Framework Run Queries" is described below

© 2018, IJCSE All Rights Reserved

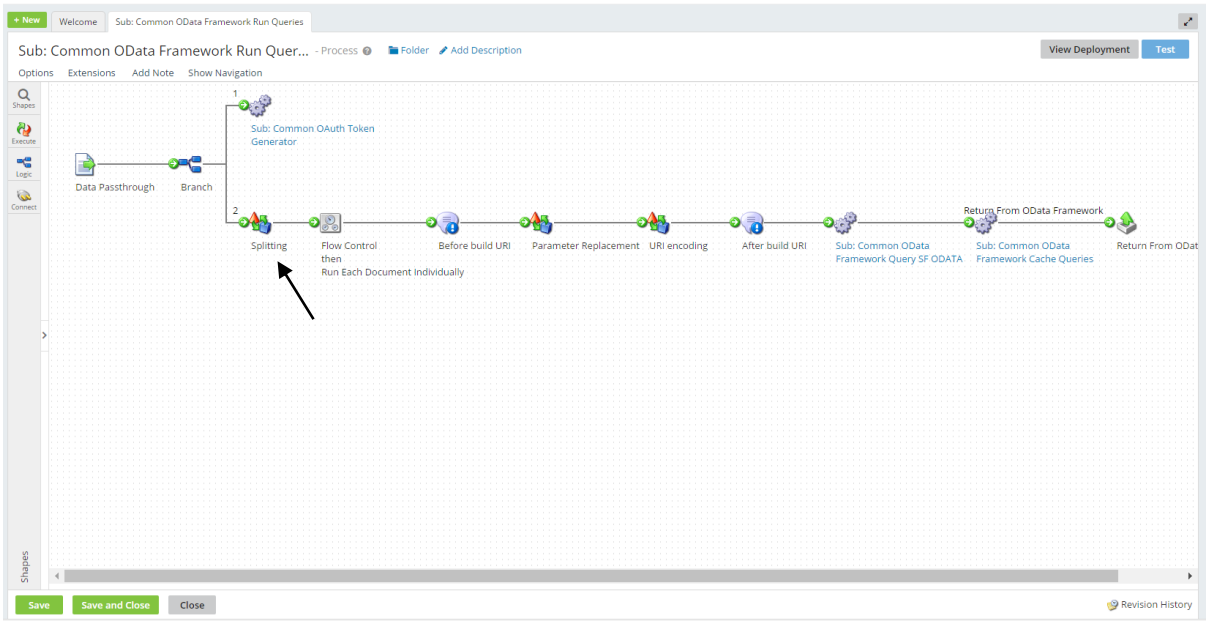
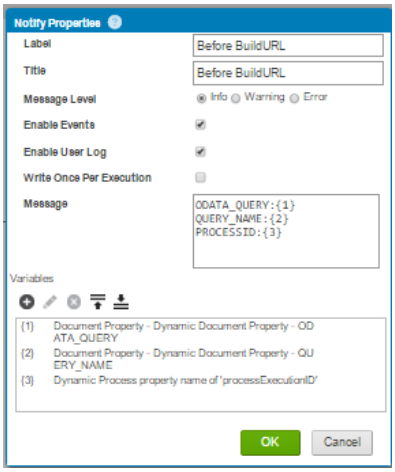


Fig 3. Second Path in Token Generation

B. Split Document logic:

Once splitting of quires is done the individual queries will be passed one by one to the notification where it keeps track of the query by registering it into a log.



For passing multiple queries, need to perform parameter replacement by using above scripting. For OData queries need to build the URI, once the parameter replacement takes place the queries are ready to send to the connector but the only thing which is missing is URI encoding that is done by using below piece of code, this code is generic for all sort type of data.

First, split the query to insert URI path and then pass this URI to SF OData by setting up the properties and then retrieving the data. The steps that are implemented in my research paper are described below

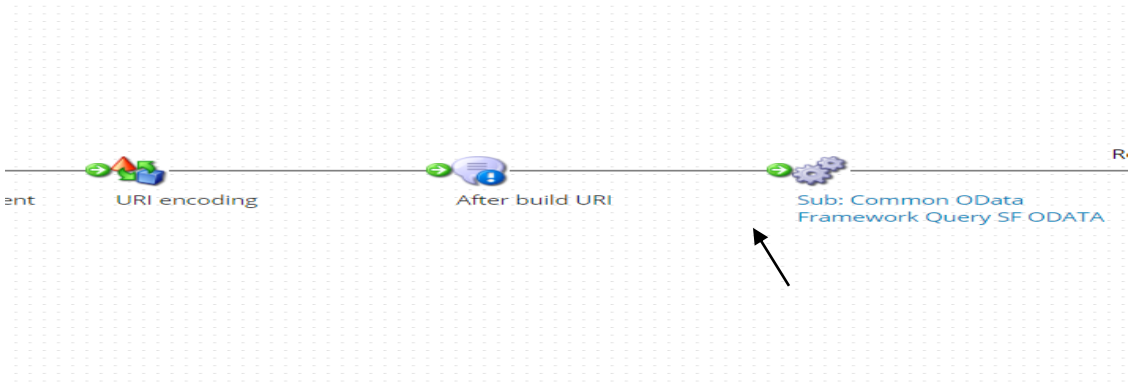
1.Parameter Replacement:

The Parameter Replacement Data process replaces the parameters in each query with the respective values.

2.URI Encoding logic (Data Process Shape):

The URI Encoding Script is used for encoding the query to form a URI , which is futher sent as a http request to get the desired response.as per the below screen shot after building the URI the URI is Sent to the sub process called “Sub: Common OData Framework Query SF ODATA”

© 2018, IJCSE All Rights Reserved



From the "Sub: Common OData Framework Query SF ODATA" the URI further passes through the sub process called "Sub: Common OData Framework Call SFSF"

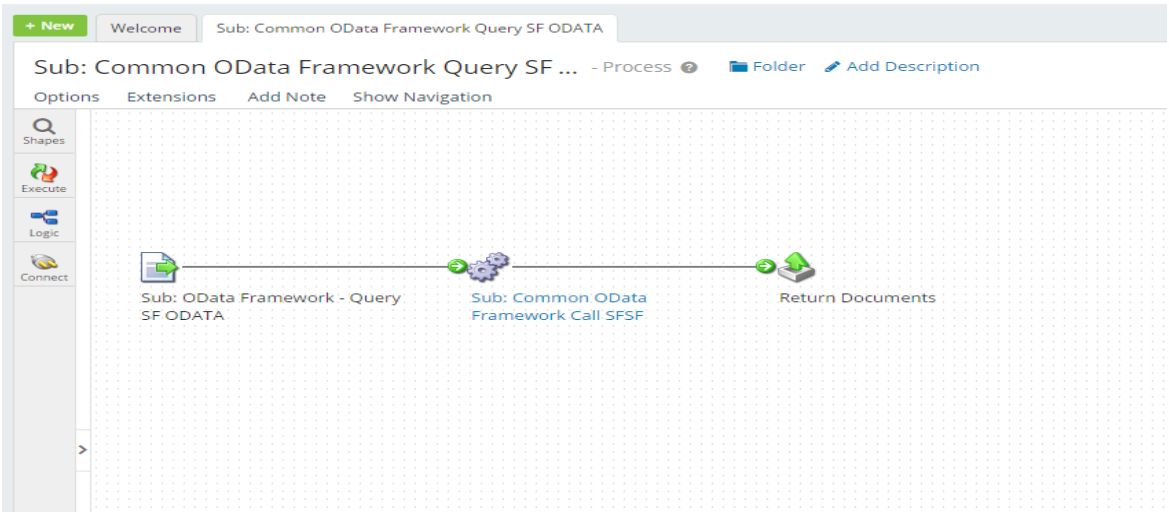


Fig 4. SF OData Query

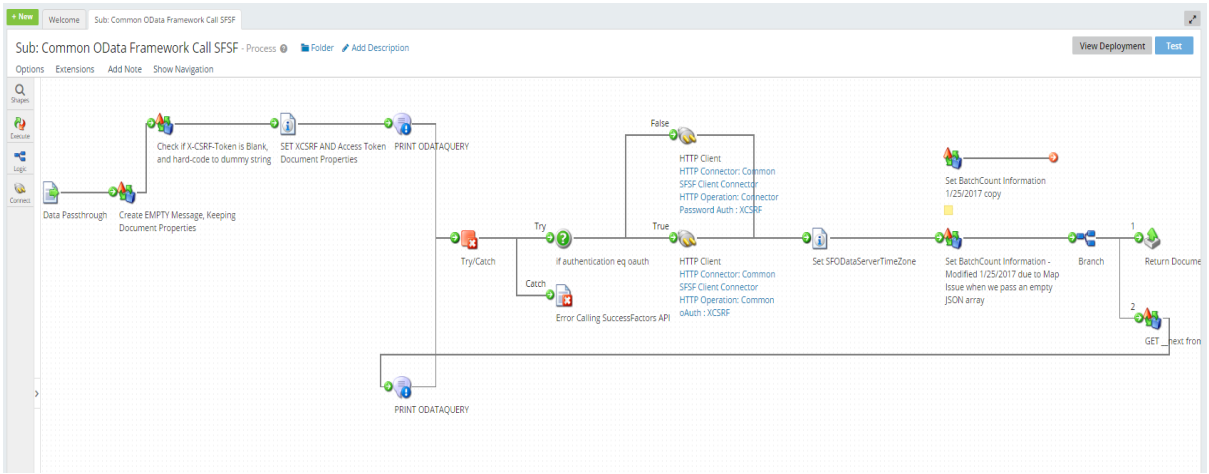


Fig 5: OData Framework - Call SFSF OData API

© 2018, IJCSE All Rights Reserved

All the OData queries are then notified to keep a log during the run and then making a query to the source system using HTTP connector. It can be seen in above screenshot, and response to the query will be returned to the Process Call that is Sub: Common OData Framework Run Queries.

In the given Fig 5, there is decision shape named as "if authentication eq OAuth?". If the value from interface concur properties which are initially at the Extensions equal to OAuth, it leads to true path otherwise leads to false paths, like as shown in the above figure. For every after 1000 records in a response document of the query,an X-CSRF token is created in the form of encrypted format. Here the X-CSRF token is used to query the next 1000 records, in a single document response which will be created in the form of json format, as per request in the Http connector.

Using this X-CSRF token, navigate the next document. And the loop continues till the end of the data from the SuccessFactors. For every loop the document is passed through the data process called "Set batch count information" as shown in the Fig 5. The two HTTP connectors which are in Fig 5 has two different operations. If the decision is true it creates a response of the query using the OAuth authentication process. otherwise, there will be a response created for the query with the basic authentication.

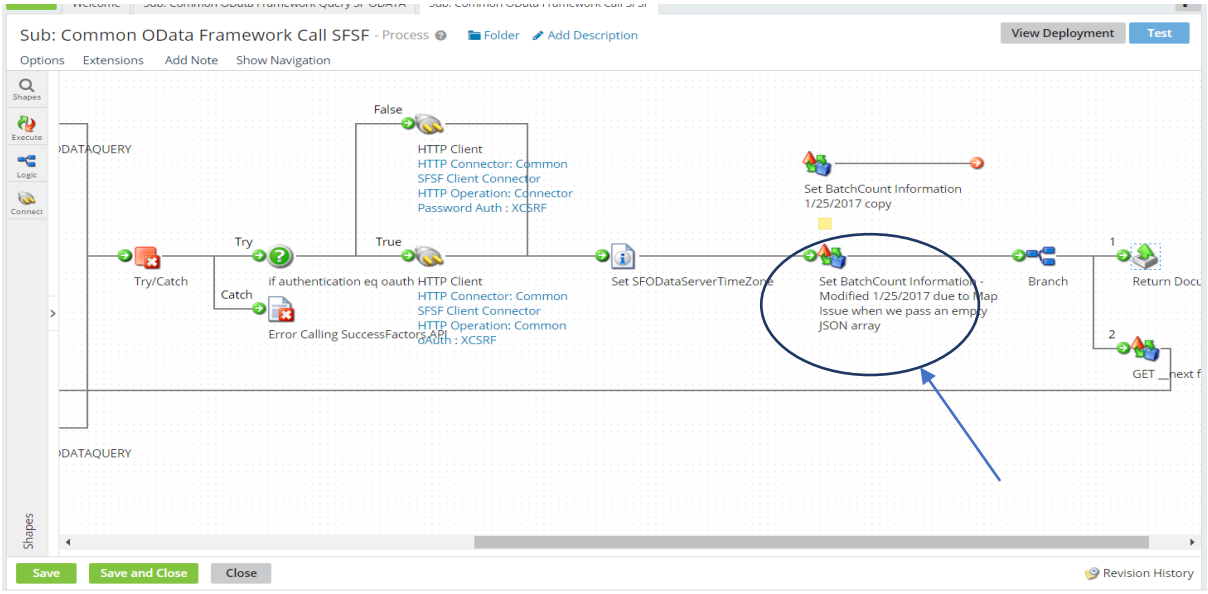


Fig 6 : Common OData Framework Call SFSF

To retrieve the data from SF using HTTP connector after authentication mechanism, batch procedure is used to maintain count of batches that are to be pulled from SF. The batch procedure includes two steps. They are

1.**Get Batch count**

Above logic is used to generate the batch count for the response of all the query made to HTTP connector. For every document in the corresponding query result, it creates a batch Number and the Batch number is going to set as a document property for each document.

2.**Set Batch Count data Process**

Entire documents will be going to be divided into the batches. Based on the batch count again they are combined into one node in the final employee profile. All these are used to overcome the java heap space error.

After completing all the query response all the documents are returned to the sub process called "Sub: Common OData Framework Run Queries" as shown in the Fig 7.

© 2018, IJCSE All Rights Reserved

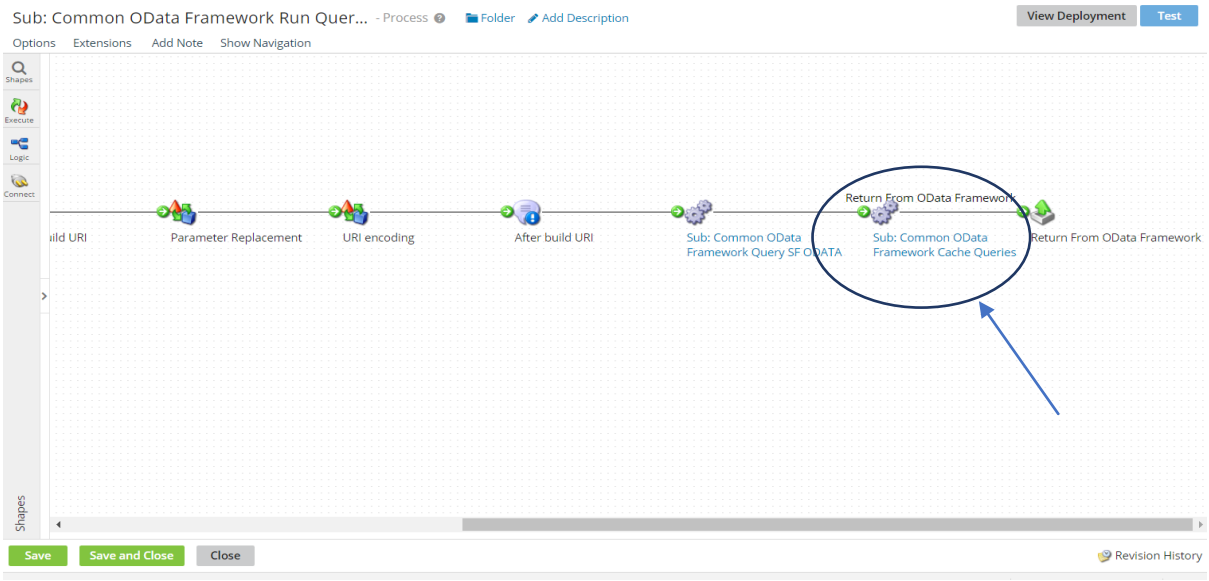


Fig 7: Sub Process call

And All the returned documents will be sent to the document cache as shown in the below figure.

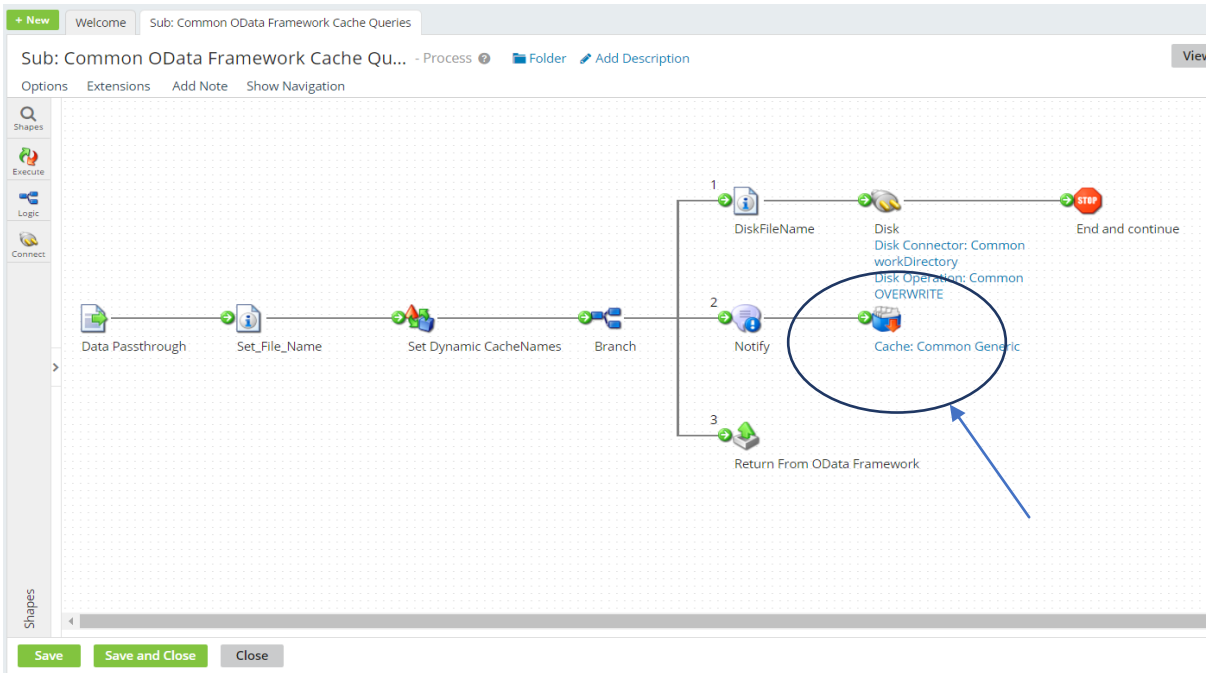


Fig 8 : Intermediate results to cache

Now till this point of the process, the query is created, that query is sent to get the response which is specific to one type of data. The return document will be passed to the called process.
Going back to the called process Sub: Common OData Framework Run Queries, next process call is 'Sub: Common OData Framework Cache Queries'.

IV. RESULTS

The proposed method could provide very secured authentication and authorization. The framework that we have used in our research work has integrated the users of cloud and thus data was retrieved in a secured and efficient manner. Back up to the data

© 2018, IJCSE All Rights Reserved

was provided to make the retrieval loss less. We have stored the results that are pulled from the SuccessFactors in cache and in cloud.

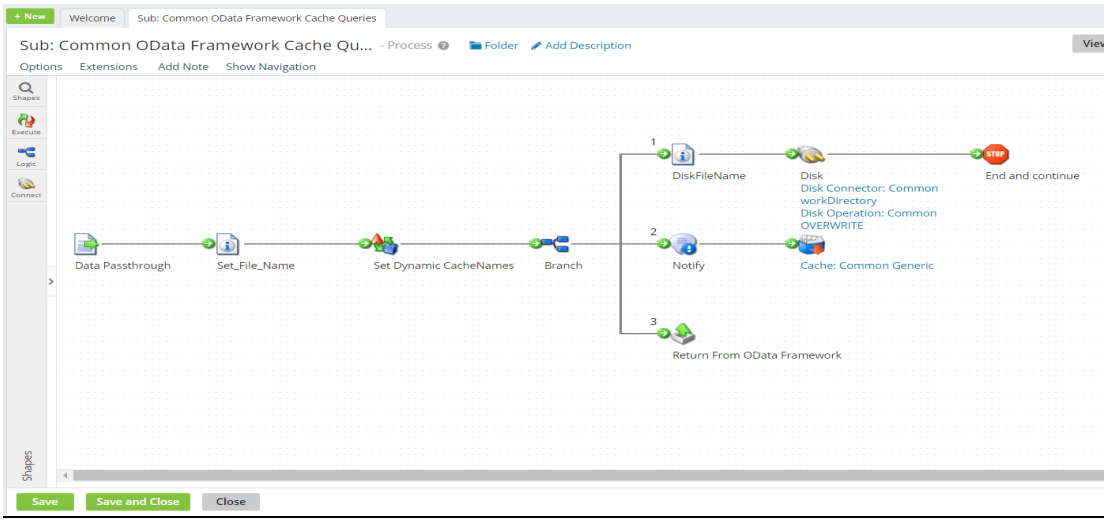


Fig 9: Common OData Framework Cache Queries

In Set_File_name 'Set property Shape', we are setting properties called Set TZ Variable and file name for dynamic cache names, this property is set to give a name to the response files that are going to be saved in SFTP or Atom directory. To set up the file name we have set the Dynamic Process Property where the file name is going to be the concatenation of process Execution ID and query name as per the given screenshot in Fig 9.

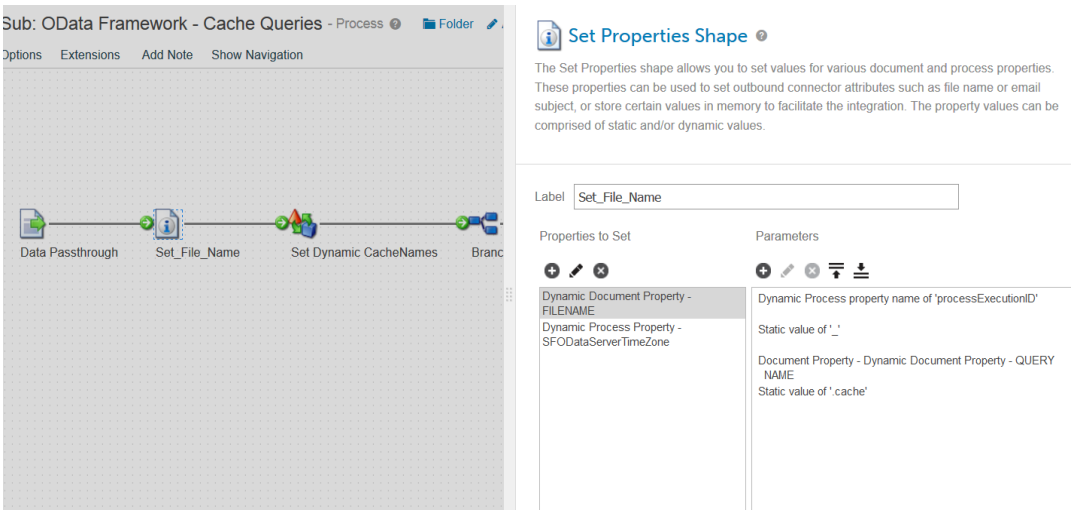


Fig 10 : Set Properties

By adopting the following procedure, the file will be finally retrieved from the cloud. Cache is the temporary storage place. The File name will be generated in the below format 'Process ExecutionId_Query_Name Cache'. And set the time zone as 'SFODataServerTimeZone'. Setting Cache names, for example Address Cache, Address details, Phone cache for phone details, Personal Cache for the personal information...etc. is important for the creation of Employee Profile. After the above process we have sent these Cache names to the Disk _Temporary_cache_Folder i.e., Atom Working Directory. 'Notify' shape is used to display the logger message.Whenever an error occurs it shows the log as 'Add To Generic Cache'. After that, there is an 'Add to Cache'shape which is named as 'Cache: Common Generic'.By using this we have added the documents to the'Generic

© 2018, IJCSE All Rights Reserved

cache'. Here Query Names are added as Document Property.And in the last path all the documents are returned to "Sub: Create Employee Profile".

V. CONCLUSION

Using Dell Boomi integration tool along with OAuth authentication process files can be retrieved from the cloud in highly secured manner. Before generating queries to the database for information retrieval authentication is being checked by using asymmetric key technique and token generation. With this method for every session authentication and authorization of the client is being checked by the third party. Token is generated to maintain timestamp for the session. With our proposed method we were able to provide high level of security to retrieve required information from the cloud. By using middleware integration tool, we have established reliable and secure connections between the nodes that were accessing the cloud.

Our research aims to solve the security control and authorization issues encountered while accessing the data from the cloud. This paper refers to open standards like OData API and OAuth to solve the problem by account authentication, access right authorization and secure retrieval mechanism. This research work also demonstrates secure cloud integration that is capable to manage and synchronize file retrievals and data access. The APIs used in this research work could provide secure integrations between the nodes that use the cloud for CRUD style access.

VI. REFERENCES

[1]. Johannes K. Chiang, Eric H.W. Yen, Yen-Hua Chun, Authentication, Authorization and File Synchronization on Hybrid Cloud on Case of Google Docs, Hadoop, And Linux Local Hosts, IEEE DOI 10.1109/ISBAST,2013

[2]. Ja-Hwa Liu, Shi-Kai Huang: "A Research into Protection Mechanism for Cloud Information Security", Proc. of the 2010 Conference on Computer Vision, Image Processing and Information, Zhongli Taiwan, Jun. 9, 2010

[3]. Liang-Jie Zhang, Quan Zhou, "CCOA: Cloud Computing Open Architecture," IEEE DOI 10.1109/ICWS, 2009.

[4]. Harry Katzian Jr, "On the Privacy of Cloud Computing", International Journal of Management and Information Systems. Littleton: Second Quarter 2010. Vol. 14, Iss. 2; p. 1

[5]. Bhaskar Prasad Rimal, Eunmi Choi, Ian Lumb, "A Taxonomy and Survey of Cloud Computing Systems", ncm, pp.44-51, 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009

[6]. W Michael Ryan, Christopher M Loeffler, "Insights into Cloud Computing" Intellectual Property & Technology Law Journal. Clifton: Nov 2010. Vol. 22, Iss. 11; p. 22

[7]. Qingqing Xie, Liangmin Wang and Hong Zhong "Non-Repudiable Query Answer Authentication Scheme over Anonymous Cloud Data", 2016 International Conference on Advanced Cloud and Big Data, IEEE DOI 10.1109

© 2018, IJCSE All Rights Reserved