API-led Connectivity Using Mulesoft For Healthcare

Mayur Raj Singh Chouhan[1], SujataPriyambada Mishra[2]

[1]Student, Dept of Electronics and Communication, R.V. College of Engineering, Bangalore, INDIA -560102

[2]Professor, Dept of Electronics and Communication, R.V. College of Engineering, Bangalore, INDIA -560102

[1]mayurair01@gmail.com[2]sujatapm@rvce.edu.in

Abstract:*API-led business model is a new way of thinking about how to engage with partners and customers through APIs. The traditional approaches were designed for business use cases having limited endpoints and speed of delivery is not critical. Traditional approaches include P2P and E2E.To overcome these drawbacks, a new approach of API-led connectivity is used. The primary aim of API-led connectivity is to allow the integration flows to be reused by means of many events and to be reused within the integration platform. This paper focuses on creating an API- led network which is agile, reusable and fits business requirements. This network will be built for the health care system using Allscripts and Fitbit APIs. To implement API -led networks in this paper MuleSoft platform is used and API are built using API development cycle. Depending on the business requirements on search for API and connector already present on Anypoint App exchange. Using available component flow is created to connect EHR system systems and different devices like Fitbit. Both API are protected with OAuth 2.0 which is handled with separate flow for initiating the communication between the system and source API. After doing the required transformation to data fetch from source API data is pushed to the Salesforce object. Using different services provided by Mulesoft, an API led network was built in which EHR and the Fitbit device were integrated. As this development process followed agile behavior it helped in meeting business requirements for building networks. The data was successfully fetched, processed and transferred form EHR system and the Fitbit device to the Salesforce object.*

Keywords:**MuleSoft, Salesforce, Anypoint Studio, EHR, API-led connectivity**

1. Introduction:

In today's age of APIs, they are no longer just a technical interface on top of a database. Nowadays API represents a new business model. Scope of API is no longer limited to internal use, they are exposed by a developer for external use cases globally. For example, Uber APIs are used by Google Maps to predict travel fares by different taxi services. This API-led business model is a new way of thinking about how to engage with partners and customers through APIs. In other words, the product is the API, so one must take proper care to design it based on business criteria, deploying and managing the APIs. The traditional approaches were designed for business use cases having limited endpoints and speed of delivery is not critical. Traditional approaches include P2P and E2E. The P2P approach connects one business operation to another operation using a direct connection.

For a use case where a lot of applications are integrated, it becomes very inefficient. The main drawback includes high market time, maintainability and hard to change. E2E methodology focuses on incorporating data however much as could reasonably be expected. In this methodology, an incorporation stage is utilized ESB, which fills in as the base for gathering all the data and serving them to the last recipient. It concentrates and reuses segments like logging, blunder taking care of, exchanges, and so on This methodology is substantially more proficient than the P2P approach. Nonetheless, to meet the computerized change of today's age, it is as yet not proficient enough, because "time to market" is still more with this methodology. To overcome these drawbacks, a new approach of API-led connectivity is used.

The primary aim of API-led connectivity is to allow the integration flows to be reused by means of many events and to be reused within the integration platform. With the re-usability of the already available algorithm (implemented in flows), the developer can improve their algorithms in more secure and quicker ways, leading to a quick time to marketplace. APIs are created in layers and the quality plus factor in comparison to E2E method is that more additives (flows) can be reused which makes it simpler to enforce new systems and offerings. For this approach, APIs are divided into three categories. Fig 1 show example for API-led connectivity

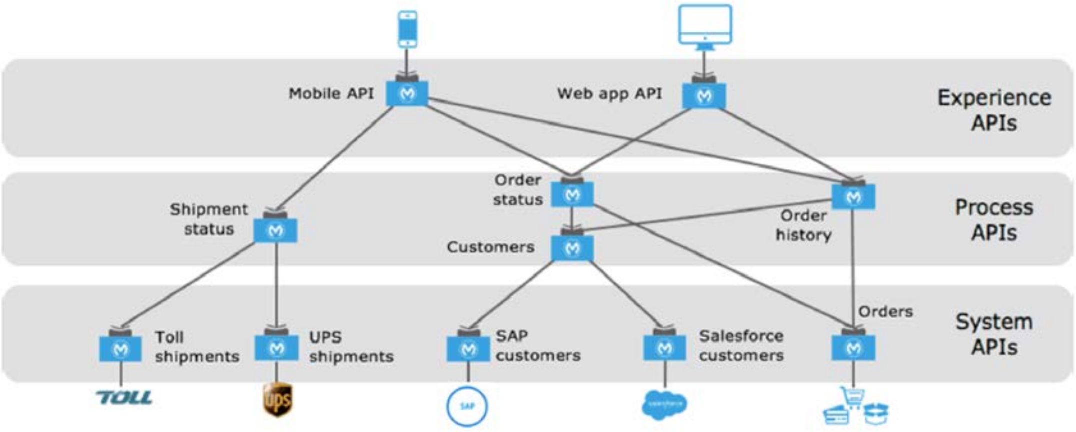

Figure 1 API-led network

System API are commonly access the core structures of report and provide a method of insulating the user from the complexity or any modifications to the underlying structures. As soon as constructed, many customers, can get right of entry to facts without any need to analyse the underlying structures and may reuse these APIs in a couple of Process APIs engage with and shape information within a single system or across systems and are created here without a dependence at the source systems from which that information originates, in addition to the target channels via which that information is delivered.Experience APIs are the means by which information can be reconfigured so that it can be easily used by the intended target market, all from a common information source, rather than building separate point-to-point integrations for every channel. An experience API is generally created with API-first design principles in which the API is designed for the specific user experience.

2. Proposed methodology:

For developing API development is used API development cycle is used. Anypoint platform, with it using philosophy of API-led connectivity, follows a similar development life cycle to software development cycle: Engage, Design, Develop and Test, Deploy and Operate. Fig 2 shows the development cycle wheel for API.

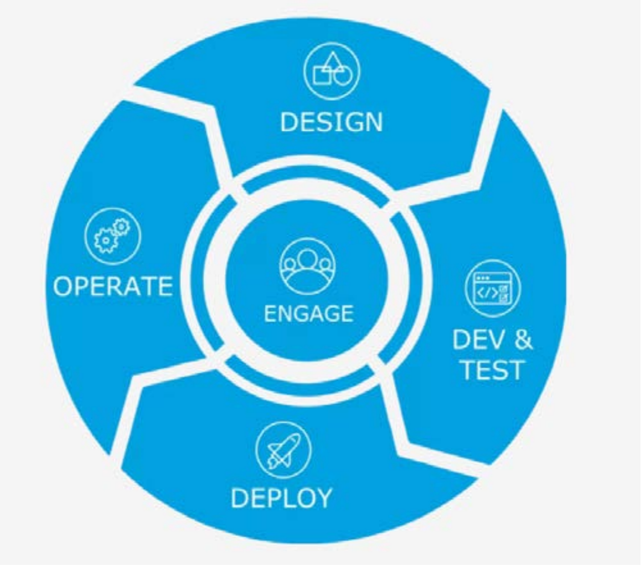

Figure 2 API-development cycle

2.1. Engage

Mulesoft has the Anypoint exchange to avoid reinventing the wheel. It is a public repository of templates, connectors and API. Using present assets on exchange time required for building a network can be reduced significantly. For All script and Fitbit there are templates available for reference.

2.2. Design, Develop, and Test

With the help of Anypoint Design center APIs can be designed, developed and tested. It includes tools such as Anypoint Studio, Flow designer and API designer. Using these tools, developers and Integration Trailblazers can:

- Utilizing online graphical interface design,document and test APIs.
- Utilize open API specifications to scaffold the mock (simulate) an API, automatically generate documentation, or API implementations.
- Build or prototype integration's and implement APIs in a low-code environment in browser or desktop.
- Create data transformations by writing the code or within a drag-and-drop interface.

1.3. Deploy and Operate

Anypoint management center allows to execute following feature:

- Deploy, monitor APIs, and manage and integration on-premises or in the cloud.
- Provision access across entire teams or individuals with single sign-on technologies such as SAML and OAuth.
- Manage proxy, control versions, apply pre-built or custom policies, and set alerts.
- Track key metrics like API usage, transactions by region, and performance with pre-built or custom dashboards.

2.4. Selection of RESTful APIs

Allscripts and Fitbit both offer various APIs to fetch different types of data. Mapping is needed to be done between availability and business requirements.Once APIs are selected with the help of sample response API metadata should be created in json format.

- API for Allscripts: Patient and Allergy intolerance are used because this provides all required details and patients personal information and medication.
- API for Fitbit: Get Activity and Get Food log are used as they provide input for daily activities details like calories intake, calories burnt, water intake etc.

After selecting API after creating metadata for input, objects with similar metadata should be created in Salesforce org which is the final destination of data.

3. Implementation:

3.1. EHR integration using All Scripts

Allscripts healthcare answers, is a publicly traded American agency that gives health practitioner practices, hospitals, and other healthcare carriers with practice management and digital fitness record generation.EHR data can be fetched using Allscripts API.

3.1.1. Registration of All Script app: To use all script API, firstly apps need to be registered on their developer’s website "https://developer.Allscripts.com/". Create a FHIR application using following details:

- App Name: Fitbit App
- RedirectURLs: https://localhost:8081/callback
- Client type: Confidential
- App type: web app

After saving the application, the portal will generate a client ID and Client secret key for the application, save this information for later use.

3.1.2. Implementing All Script on MuleSoft

Implementation of EHR integration is done using the RESTful API of Allscripts and Anypoint studio. Request, transform, create module from the Salesforce library are the key modules used to fetch data from Allscript,process data and push data to Salesforce.

3.1.3 Authorization of Allscripts API in Anypoint studio: AllscriptsRESTful APIs are used to connect with Anypoint studio using Request module. Connection is established after OAuth completion. For configuring OAuth authentication add OAuth module dependencies in pom.xml file of paper and configure request module as follows-

- Keep authentication “authorization code grant type”.
- Fill Authorization URL, Token URL client id client Secret, scope.
- Keep external call URL same as redirect URL registered in APP.

- Create local call back configuration with default settings.
- Add the callback path as "/callback".
- Local authorization URL will be used to get and store access tokens.
- Keep all settings as default.

After the Request module is configured with OAuth settings, a different RESTful API can be used to access different patient data from Allscripts.

3.1.4. Processing data and Deployment of data: Each Allscripts API call will require parallel flow to fetch the data and transfer it to Salesforce. Once data is fetched from EHR, it needs to be filtered as per the business requirement. Using the metadata from required output creating a mapping function.

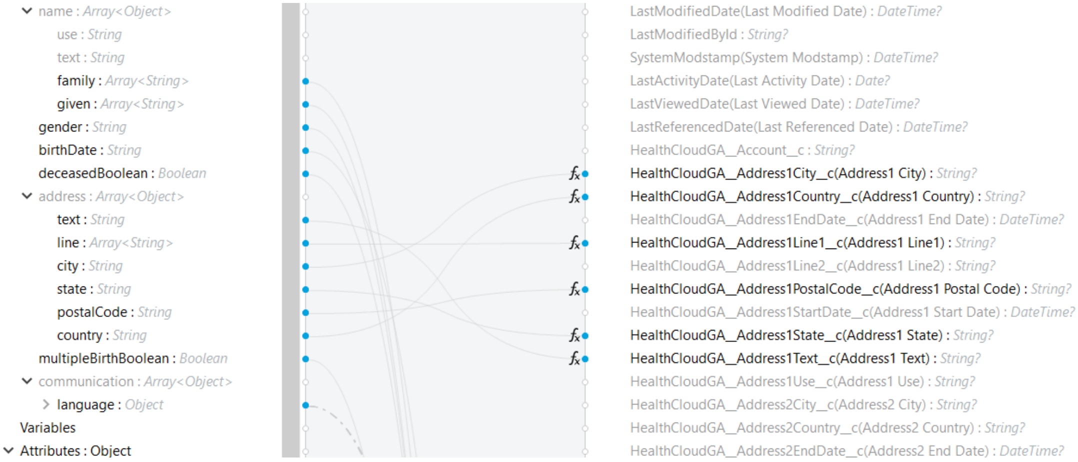
name : Array<Object>
use : String
text : String
family : Array<String>
given : Array<String>
gender : String
birthDate : String
deceasedBoolean : Boolean
address : Array<Object>
text : String
line : Array<String>
city : String
state : String
postalCode : String
country : String
multipleBirthBoolean : Boolean
communication : Array<Object>
language : Object
Variables
Attributes : Object
LastModifiedDate(Last Modified Date) : DateTime?
LastModifiedById : String?
SystemModstamp(System Modstamp) : DateTime?
LastActivityDate(Last Activity Date) : Date?
LastViewedDate(Last Viewed Date) : DateTime?
LastReferencedDate(Last Referenced Date) : DateTime?
HealthCloudGA__Account__c : String?
HealthCloudGA__Address1City__c(Address1 City) : String?
HealthCloudGA__Address1Country__c(Address1 Country) : String?
HealthCloudGA__Address1EndDate__c(Address1 End Date) : DateTime?
HealthCloudGA__Address1Line1__c(Address1 Line1) : String?
HealthCloudGA__Address1Line2__c(Address1 Line2) : String?
HealthCloudGA__Address1PostalCode__c(Address1 Postal Code) : String?
HealthCloudGA__Address1StartDate__c(Address1 Start Date) : DateTime?
HealthCloudGA__Address1State__c(Address1 State) : String?
HealthCloudGA__Address1Text__c(Address1 Text) : String?
HealthCloudGA__Address1Use__c(Address1 Use) : String?
HealthCloudGA__Address2City__c(Address2 City) : String?
HealthCloudGA__Address2Country__c(Address2 Country) : String?
HealthCloudGA__Address2EndDate__c(Address2 End Date) : DateTime?

Figure 2 Mapping functionality

Fig 2 shows a subset of complete mapping in which fields received from patient’s data are mapped fields of Salesforce fields. For transferring data "create module of Salesforce library is used, which creates a single record in particular object of Salesforce.

3.2. Fitbit Integration using Mulesoft

These gadgets are utilized to follow diverse wellbeing data of clients like advances pulse oxygen level in blood, and so forth.

3.2.1. Registration of Fitbit app: Similar to Allscripts, apps need to be registered on “https://dev.fitbit.com/apps/new” for using the Fitbit API. After creating an open Fitbit account and allowing access to use data.

3.2.2. Implementation on Anypoint studio: For creating complete functionality like registration, authentication getting observation using Fitbit API several flows are created. Following flows are required to implement complete functionality of Fitbit integration:

- To register a user with Fitbit: This flow takes access code and URI parameter patient ID, then sends a get request to fetch access token, refresh token and user ID and transfer control to save token flow.
- To save access token: Flow is used to store and log userID, access token, refreshtoken, client ID into persistent objects.

- To refresh access token: As OAuth 2 requirement access token is required to refresh after a fixed period of time, this flow is scheduled to run after completion of about 50\% of expiration time. After fetching new access token control is transferred to save access token flow and previous values are replaced with new values.
- To get observation: This flow is used to fetch observational data using the Fitbit API with the help of a request module creating a GET request. After fetching data is mapped with a Salesforce object.
- To remove user: This flow is used to remove patient information saved in cache

4. Results:

Data stored in Fitbit was successfully fetched, processed and transferred to Salesforce org. Get activity and get food log were used to fetch data from Fitbit device and transfer it to the Salesforce org. Following are the flows created to integrate Fitbit.

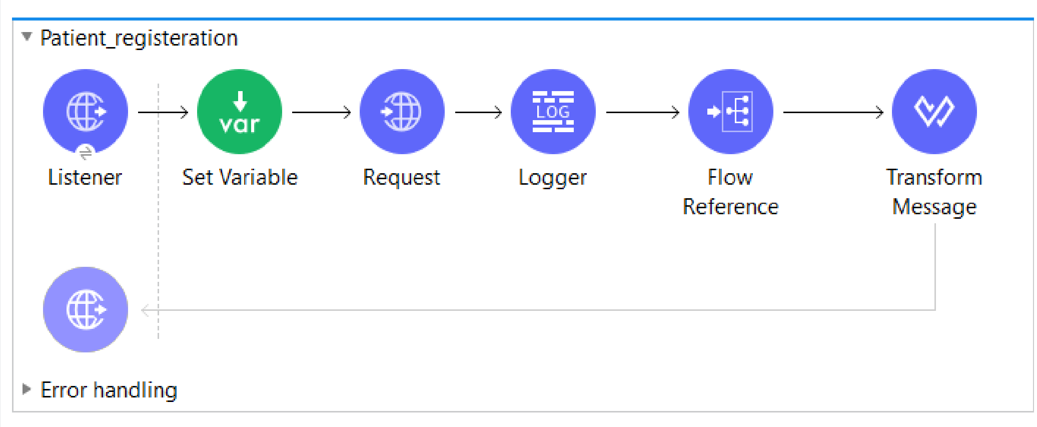

Figure 3 Patient Registration Flow

Different flow has different functionalities flow in Fig 3 is used to takes access code and URI parameter patient ID, then send a get request to fetch access token, refresh token and user ID and transfer control to save token flow,

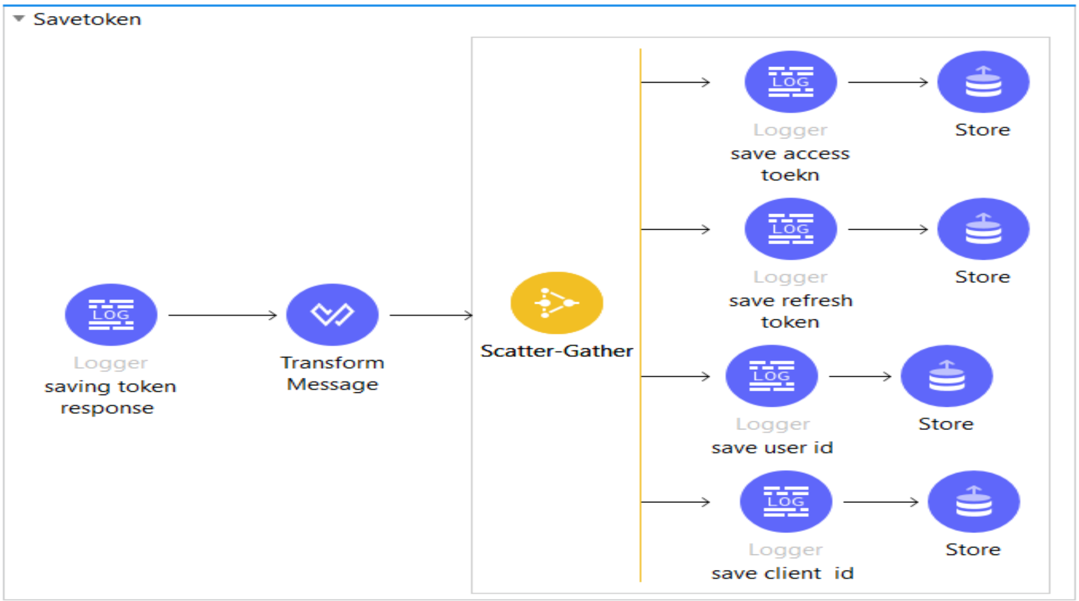

Figure 4 User information saving flow

Flow in Fig 4 is used to store and log userID , access token ,refresh token , client ID into persistent objects.

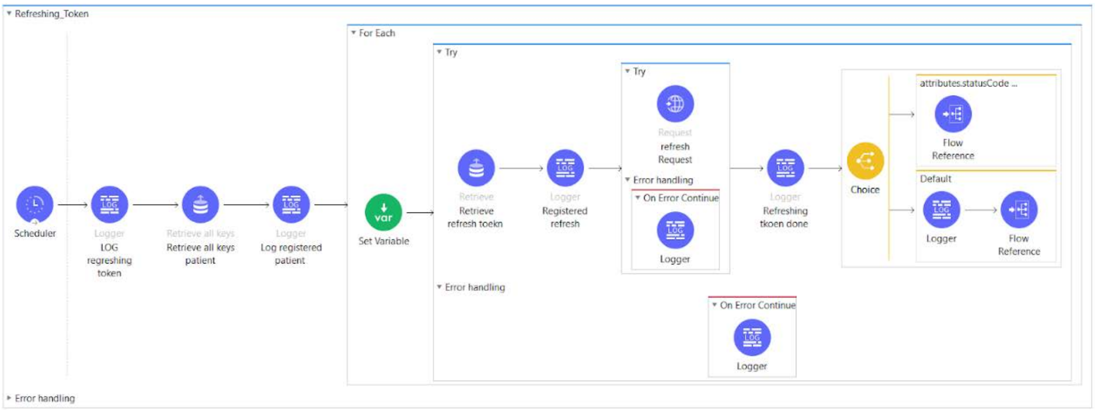

Figure 5 Flow for refreshing access token

Flow in Fig 5 is used to refresh the access token of each patient.

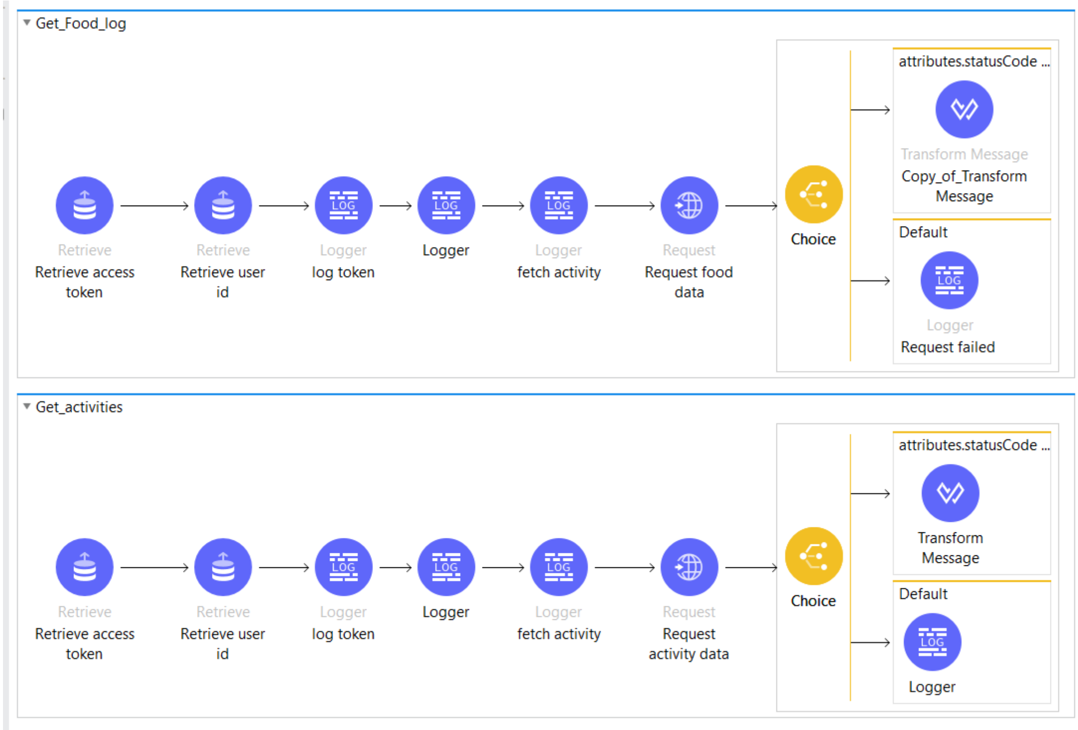

Figure 6 Flow for fetching observation from Fitbit

Flow in Fig 6 is used to fetch observational data using the Fitbit API with the help of a request module creating a GET request. After fetching data is mapped with a Salesforce object.

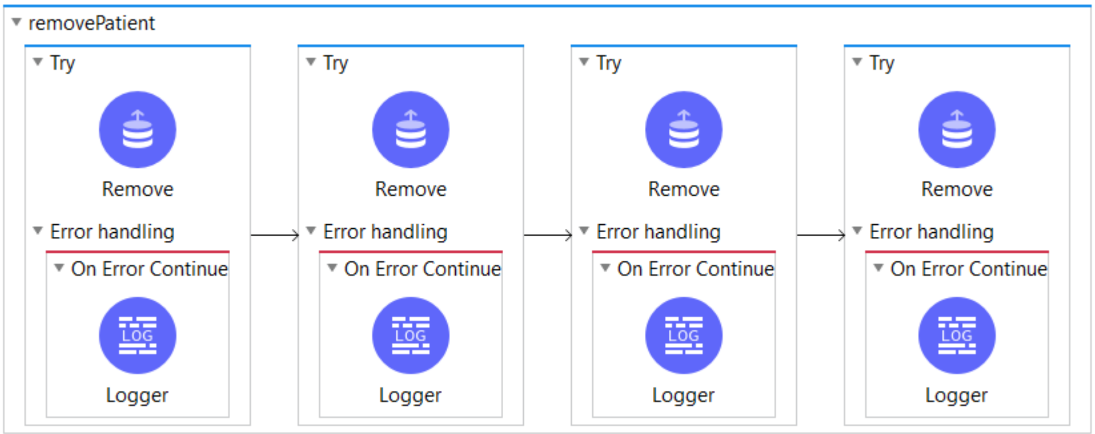

Figure 7 Flow for removing patient information

Flow in Fig 7 is used to remove patient information saved in cache. These all combined are used to create Fitbit API.

4.2. Allscripts results:

With the help of Salesforce connector Allscripts API and default module of Mulesoft flow was created to fetch and transfer data to Salesforce as shown in Fig 8.

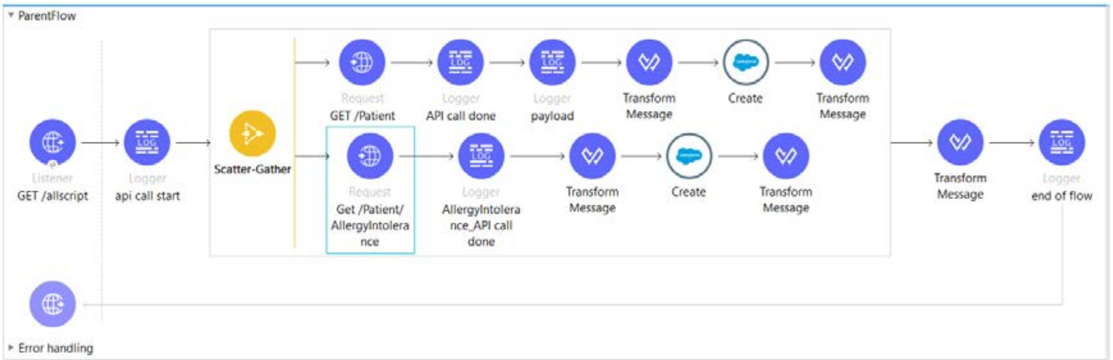

Figure 8 Flow for patient record

Data from the EHR system was successfully fetched and transformed as business requirements. API was tested using Postman and gave a successful response. Data was successfully transferred to the Salesforce object.

5. Conclusion and Future Scope

API-led connectivity can help businesses scale very fast with its agile behavior. In this paper API network was created for health care systems using integration of EHR data and a Fitbit device. Data received from a corresponding source requires process and filtering after which data can be transformed to the target which in this case is Salesforce. With its agile and connected philosophy Mulesoft played a significant role in making integration agile and scalable. For creation of API for Allscripts and Fitbit API development cycle was implemented which reduced time to build the system as reinvention of the wheel for certain components was not required. For EHR integration, Allscripts API was used to fetch data from EHR and then processed using Mulesoft Anypoint Studio. After which data was transferred to the Salesforce object. For Fitbit integration Fitbit APIs were used to fetch data and process it. Which gave insight about the routine of Fitbit owner. With the help of a combination of data fetched from All scripts and the Fitbit API it can be used in the Healthcare system to create medication plans and have more insightful observations over a particular patient.

6. References:

[1] M. Seth, "Mulesoft – Salesforce Integration Using Batch Processing," 2018 5th International Conference on Computational Science/ Intelligence and Applied Informatics (CSII), 2018, pp. 7-14, doi: 10.1109/CSII.2018.00009.

[2] L. Li and W. Chou, "Design and Describe REST API without Violating REST: A Petri Net Based Approach," 2011 IEEE International Conference on Web Services, 2011, pp. 508-515, doi: 10.1109/ICWS.2011.54.Si-Jung Kim, Do- Eun Cho., Technology Trends for UX/UI of Smart Contents. The Korea Contents Association Review, 14(2016), 29-33.

[3] W. Zhou, L. Li, M. Luo and W. Chou, "REST API Design Patterns for SDN Northbound API," 2014 28th International Conference on Advanced Information Networking and Applications Workshops, 2014, pp. 358-365, doi: 10.1109/WAINA.2014.153.
[4] Fitbit,' Web API'. [Online].Accessed on: May19, 2021. Available: https://dev.fitbit.com/build/reference/web-api/
[5] Allscripts, 'FHIR-GETSTARTED' [Online]. Accessed on: May 25, 2021.Available:https://developer.allscripts.com/FHIR?PageTitle=GetStarted.
[6] MuleSoft 'MuleSoft Documentation' [Online]. Accessed on June 5,2021.Available: https://docs.mulesoft.com/general/.