Reusable Integration Health Check Framework using IIB

Document Control

Title	Reusable Integration Health Check Framework using IIB
Author	Umasankar Gnanasekaran
Doc Ref	NA

Owner (Responsible for Approval of Issued Versions)				
Name	**Role**	**Signature**	**Date**	**Issue**
Umasankar Gnanasekaran	Integration Architect	Digital signature	DD/MM/YYYY	

Review Panel			
Name	**Role**	**Name**	**Role**

Table of Contents

1. Introduction

This document has been prepared by IBM and provides an explanation of how the integration Health Check Framework is designed using IIB to check the various integration services. Also, this document describes the automation of testing of Webservice based middleware applications.

2. Scope of the Document

The scope of the framework is developed only to test the webservices and REST based APIs. The document describes the current system architecture, developed framework architecture, involved components, and technical configuration detail to re-use the framework in anywhere in the integration landscape.

3. About this Framework

The framework mainly developed to help the development or testing team to prioritize the work based on the back-end service availability. Apparently, the front-end applications always connect the integration layer to connect its backend or any other partner systems by webservice. The framework tests all the integrated services and provide a detailed excel report of each service working status by Email to the stakeholder before starting the work. Also, it sends SMS alerts to the application team if any one of the application or services not working. This framework helps the project team to increase the productivity.

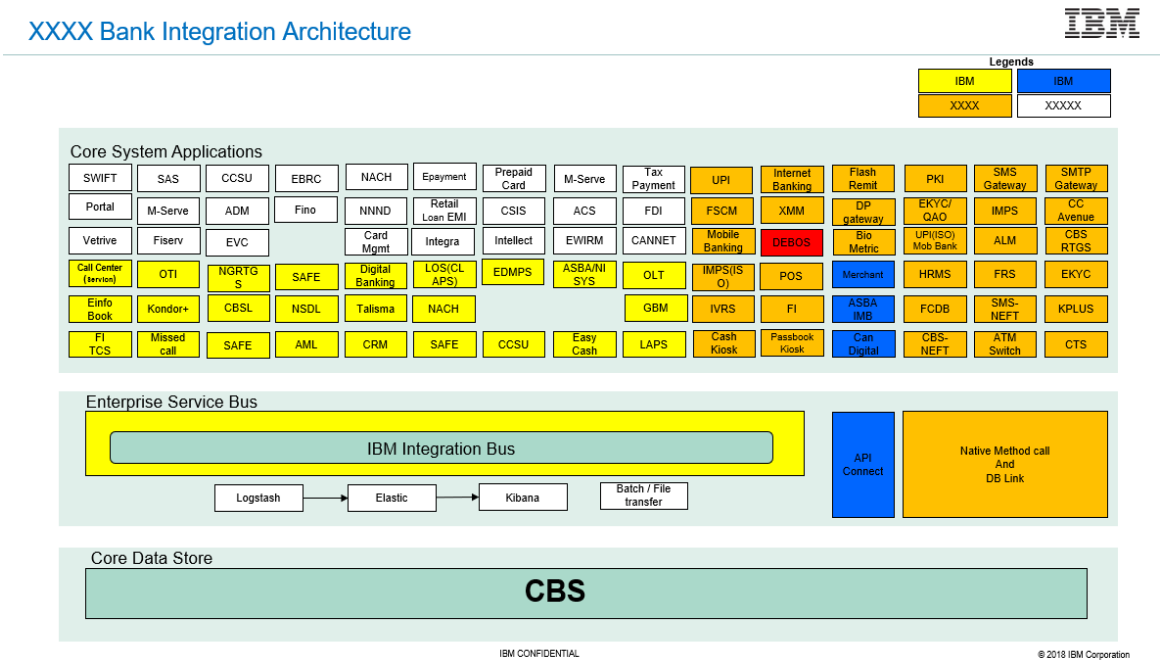
4. Key Features of the Framework

There are many key features available in the framework and some of them listed here to understand about the developed framework.

1. It solves a common problem of back end service availability to the front-end applications.
2. A generic framework can be reused in any integration project
3. It reduces the involved stake holders time & effort and they can focus on other critical work. Reduce or eliminate the manual testing effort to understand the back-end service status.
4. You can limit how many services can be tested by changing the input configuration.
5. Email report facility to the stakeholders for pre planning of BAU activities.
6. SMS alert to the application team to bring up the services for other applications.
7. It gives information about the performance of each services.
8. Familiar maintenance and very limited configurations when we move to other environments.
9. n-Number of services can be added to the framework for testing.

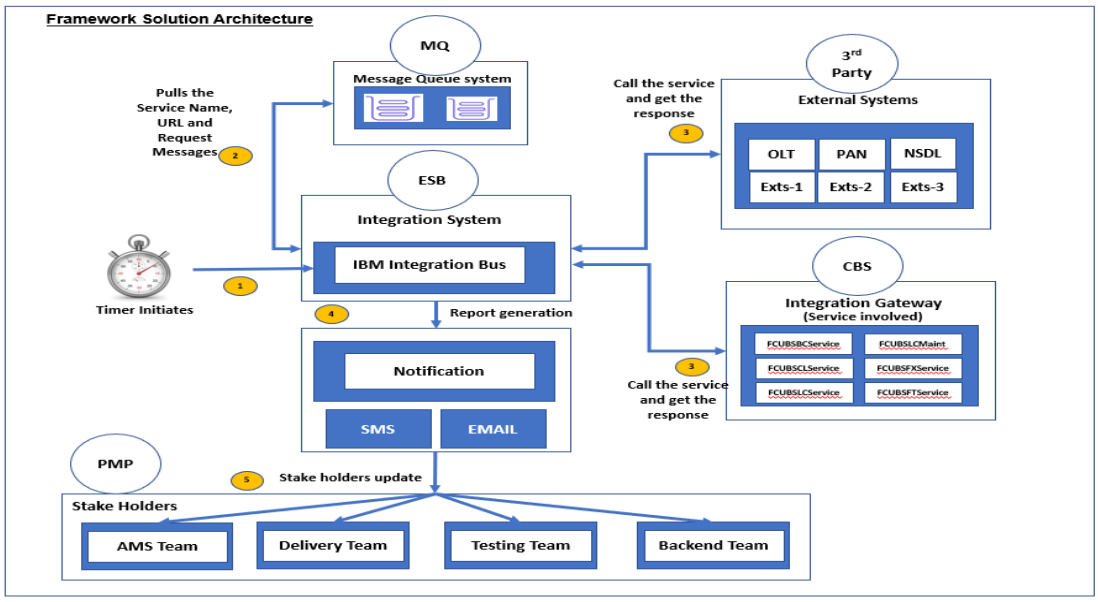
5. System Architecture

This framework has been implemented in the below mentioned solution architecture and it was executed in one of the leading banks in India.



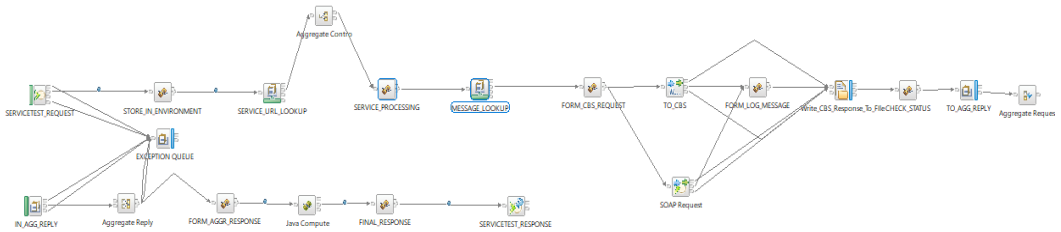
6. Framework Solution Architecture

It is a generic framework which can be used any of the integration platform. The below solution architecture designed based on the current system architecture. This architecture helps to understand the backend services availability. All the components which available part of the current implemented solution are and didn't use external component or any other licensed software to prepare the framework.



1. Service Design Approach

To implement the framework, we have used various nodes and components to achieve the solution. Below is the IIB flow design for reference.



2. Architecture Rationales

Currently, the Framework developed to support XML and JSON messages. It's an ongoing process of adding new feature in this Framework based on the features backlog update.

3. Risk

No Risk identified yet.

7. Technical Details

1. Key Components Matrix

Below are the components have been used to complete the framework.

S No	Component Name	Version	Binding
1	IIB	10.0.0.13	Mandatory
2	MQ	9.0.4.0	Mandatory
3	Java	JDK1.7 or above	Mandatory
4	SMTP server detail	NA	Mandatory
5	SMS server detail	NA	Mandatory
6	Microsoft Excel	2010 or above	Mandatory
7	Windows,AIX,Linux	Any	Mandatory

2. Framework Configuration Detail

The framework has some of the listed configuration as follows

a) IIB Service Configuration

1. The generic service framework has to be deployed in the Integration server.
2. Setup the automatic timer when you need to initiate.
3. Update your server IP and port in the URL

<http://172.16.255.149:7821/CBSServicesCheck>

b) Message Queue Configuration

The below mentioned queues are used part of the configurations,

1. LOOKUP.Q – this Queue will be used to store service details. An xml file is placed in this queue which contains the URL, service names, with which the CBS or backend service is getting invoked. If any new service status needs to be verified, then just add an entry of the service details in the xml and place it again in the mentioned queue.
2. LOOKUP.REQ.XML - This queue contains all the CBS requests in xml form, with which the CBS or backend service is getting invoked.
3. LOOKUP.SMS.Q -This Queue contains the contact details of the application owner. An xml file is placed in this queue with the detail of application owner contact details like SPOC, mobile number and etc. With which the SMS is sent to the respective system SPOC in case of a service is down.

c) Email configuration

1. SMTP server detail will be used here.
2. Security identity has to be created and updated in the Email Node.

3. Sample Reports

CBS_Services_Status_
Report_2019-04-24 0

Server Info					
	Log File Size	**Log Location**	**Minimum Heap Size**	**Maximum Heap Size**	**Broker Heart Beat**
	6386byes	C:\Workout\LogFile\ESB_Log.log	16777216	536870912	Active
SL No	**Service Name**	**Operation Name**	**Status**	**RequestTime**	**ResponseTime**
1	LOS	inquireCASAAcct	Success	2019-04-24 03_08_58.658	2019-04-24 03_08_58.803
2	ASBA	inquireCasaAccountBalance	Success	2019-04-24 03_08_58.820	2019-04-24 03_08_59.066
3	SAFE	lockerRentRequest	Success	2019-04-24 03_08_59.077	2019-04-24 03_08_59.244
4	OTI	getScreeningDetails	Success	2019-04-24 03_08_59.252	2019-04-24 03_08_59.580
5	DIGIBANK	detailsofnominee	Success	2019-04-24 03_08_59.588	2019-04-24 03_08_59.717
6	OLT	OLT	Success	2019-04-24 03_08_59.726	2019-04-24 03_08_59.767
7	GBM	GBM	Failure	2019-04-24 03_08_59.775	2019-04-24 03_08_59.767
8	CRM	CRM	Failure	2019-04-24 03_08_59.796	2019-04-24 03_08_59.767
9	Account360View	Account360View	Success	2019-04-24 03_08_59.837	2019-04-24 03_09_00.268
10	AdvancedCustomerSearch	AdvancedCustomerSearch	Success	2019-04-24 03_09_00.288	2019-04-24 03_09_00.507
11	AccountMiniStatement	AccountMiniStatement	Success	2019-04-24 03_09_00.516	2019-04-24 03_09_00.752
12	customer360View	customer360View	Success	2019-04-24 03_09_00.762	2019-04-24 03_09_01.288
13	HoldFundInquiry	HoldFundInquiry	Success	2019-04-24 03_09_01.302	2019-04-24 03_09_01.575
14	FundTransferNEFTRTGS	FundTransferNEFTRTGS	Success	2019-04-24 03_09_01.586	2019-04-24 03_09_01.839

4. Message Formats (Req / Res)

a) Timer flow – sample message

```
<CBSApplication>
<TimeoutRequest>
<Action>SET</Action>
```

```
<Identifier>ServiceTest</Identifier>
<StartDate>TODAY</StartDate>
<StartTime>NOW</StartTime>
<Interval>60</Interval>
<Count>1</Count>
<IgnoreMissed>TRUE</IgnoreMissed>
<AllowOverwrite>TRUE</AllowOverwrite>
</TimeoutRequest>
<CBSApplicationRequest>
<ServiceName>LOS</ServiceName>
<ServiceName>ASBA</ServiceName>
</CBSApplicationRequest>
</CBSApplication>
```

b) SMS Lookup – sample message

```
<AppInfoList>
 <AppInfo>
  <AppName>LOS</AppName>
  <ContactPerson>Krithika</ContactPerson>
  <MobileNumber>9945654658</MobileNumber>
 </AppInfo>
 <AppInfo>
  <AppName>GBM</AppName>
  <ContactPerson>Harish</ContactPerson>
  <MobileNumber>9912354658</MobileNumber>
 </AppInfo>
</AppInfoList>
```

c) Generic Framework service – sample message (request)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:myap="http://www.example.org/MyApplicationTest">
   <soapenv:Header/>
   <soapenv:Body>
      <myap:CBSApplicationRequest>
         <myap:ServiceName>LOS</myap:ServiceName>
         <myap:ServiceName>ASBA</myap:ServiceName>
         <myap:ServiceName>SAFE</myap:ServiceName>
         <myap:ServiceName>OTI</myap:ServiceName>
         <myap:ServiceName>DIGIBANK</myap:ServiceName>
         <myap:ServiceName>OLT</myap:ServiceName>
         <myap:ServiceName>GBM</myap:ServiceName>
         <myap:ServiceName>CRM</myap:ServiceName>
         <myap:ServiceName>Account360View</myap:ServiceName>
         <myap:ServiceName>AdvancedCustomerSearch</myap:ServiceName>
         <myap:ServiceName>AccountMiniStatement</myap:ServiceName>
         <myap:ServiceName>customer360View</myap:ServiceName>
         <myap:ServiceName>HoldFundInquiry</myap:ServiceName>
         <myap:ServiceName>FundTransferNEFTRTGS</myap:ServiceName>
      </myap:CBSApplicationRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

d) Generic Framework service – sample message (response)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <NS1:CBSApplicationResponse xmlns:NS1="http://www.example.org/MyApplicationTest">
         <NS1:Service>
            <NS1:ServiceName>LOS</NS1:ServiceName>
            <NS1:Status>Success</NS1:Status>
         </NS1:Service>
```

```
        <NS1:Service>
          <NS1:ServiceName>ASBA</NS1:ServiceName>
          <NS1:Status>Success</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>SAFE</NS1:ServiceName>
          <NS1:Status>Success</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>OTI</NS1:ServiceName>
          <NS1:Status>Success</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>DIGIBANK</NS1:ServiceName>
          <NS1:Status>Success</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>OLT</NS1:ServiceName>
          <NS1:Status>Success</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>GBM</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>CRM</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>Account360View</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>AdvancedCustomerSearch</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>AccountMiniStatement</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>customer360View</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>HoldFundInquiry</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
        <NS1:Service>
          <NS1:ServiceName>FundTransferNEFTRTGS</NS1:ServiceName>
          <NS1:Status>Failure</NS1:Status>
        </NS1:Service>
      </NS1:CBSApplicationResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

8. Advantages of the Framework

1. The framework increases the reliability of your application and reduces the programming and testing effort. Since the framework has been prebuilt and tested in the client environment.
2. Framework can be upgraded by its creators, which could potentially provide new functionality, improved performance or increased stability with additional programming from your side as framework user.
3. Well written framework provides you to easily add more services to check its availability.
4. Email notification gives a clear picture of the backend system availability to the stake holders to plan their work.
5. SMS notification alerts the application team to bring up the application.

9. Customer Experience / Success story

This Framework has been created for my current project engagement and then has been made generic to be used in any project. The framework did a nice job in testing the back-end services and customers were very happy and it is helping them in prioritizing their daily activities in a well-planned manner. The framework produced nice report and SMS alert to the stakeholders, which is very valuable to customer to avoid any system outage in the development and testing phases.
Also, we received a nice feedback/suggestion from a client to improve the product further by adding more features.

10. Conclusion

One of the great benefits of the modern development landscape is the increase in system functionality. This Framework ensures a hundred percent test coverage of the backend services and ensures availability to the front-end applications. It supports to well-organized web service development process and evaluation of their performance and functionality.