ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

Deciphering Business Integration Needs with Cloud Integration

[1] Garima Srivastava, [2] Jenifer Sam
[1] Software Development Architect,[2] Software Developer
[1][2] SAP Labs India Pvt. Ltd, Bangalore.

Abstract - **With tons of cloud applications around us, integrating and consuming data from these applications has become the need of the hour. It's not very difficult anymore to build applications that operate on social networking data or even to build a flashy sentiment analysis app that reads volumes of data from the cloud. To adopt innovations in the cloud, existing enterprise solutions need to be extended or integrated with new cloud applications in various kinds of cloud platforms, primarily public cloud and enterprise/private cloud. However, such an integration becomes quite challenging as it poses its own set of questions with respect to performance, scalability and security. There are certain key integration scenarios to be considered as we strive to build bridges between various on-premise and cloud applications. This paper discusses such possible Integration scenarios and the potential methodologies that can be adopted by organizations for a seamless integration in these scenarios. Experimental study based on various implemented system shows that the proposed approaches can help industries move towards shared computer processing to achieve coherence and economy of scale. These Integration techniques that cater to enterprise needs will increase productivity and help in minimizing cost invested in on-premise maintenance.**

INTRODUCTION

Application integration is a well-known discipline combining several technologies and business domains. In order to adopt innovations in the cloud, there is a burning need to extend and integrate existing on-premise applications with new cloud applications. In addition, cloud-to-cloud integration scenarios need to be supported.

Key Integration Requirements for Enterprise Solutions:

1. Application to Application (A2A) Integration: covers the chaining of processes between applications ensuring transactional process integrity. Connectivity to third-party SaaS providers is also an integral requirement.

2. Business to Business (B2B) Integration: describes the chaining of processes between business partners. In addition to the requirements of the A2A integration, this use case requires support of a variety of industry standards and management of a potentially large number of business partners

3. Master Data Synchronization: refers to the synchronization of master data (initial load and/or delta changes) between a system of records (source) and dependent applications (target) either in near real-time or batch.

4. Data Virtualization: provides users with real-time access (read and write) to data from an external system via a standardized interface (for example, SQL), Regardless of its physical location (on-premise, private cloud, public cloud, etc.).

5. Data Integration: is purely about data exchange between systems which is not part of a well-defined business process.

INTEGRATING DIFFERENT CLOUD PLATFORMS

Described below are the various ways in which various cloud platforms and on-premise systems can be integrated to meet the above-mentioned integration requirements for enterprise needs.

1. Public Cloud Services

For solutions to seamlessly integrate into a cloud environment, it is vital that they expose their functionality as resource-based APIs. This allows convenience in accessing or replicating data. Protocols such as REST and SOAP Web Services can be used for creating such APIs.

Ideal features to be considered for choosing a protocol
•Flexibility in accessing the exact data
•Delta processing
•Asynchronous querying for optimizing data access

Recommendations

•Applications should expose APIs that expose a domain model and business functionality.
•Exposing new public technical APIs that allow direct access to internal functionality should be avoided.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

Ways of Exposing Public APIs

a. Exposure of Public API on Cloud

Public API exposure in Cloud Platform is one of the most desired ways to consume and develop applications. All open source applications provide public APIs like Facebook, Google, Twitter etc.) which can be used to create enterprise applications. One such example is the open API which can be mapped to public APIs consumption is Integrating with Maps. Many big enterprises build applications that have one or the other GIS (Geographical Information System) features which requires an integration to the Map providers (basically a public API hosted on the web).

Another such example is the open API provided by Facebook which provides a feed of user status updates and page status updates as they are posted to Facebook. Only status updates that have their privacy set to 'public' are included in the feed.

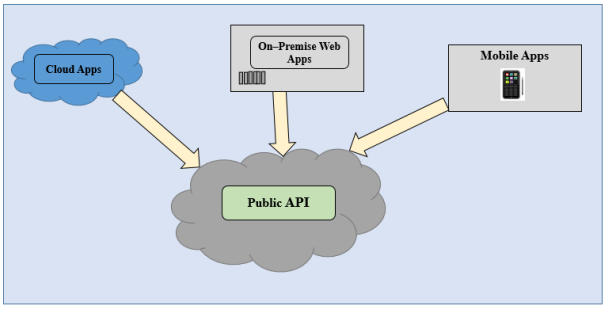

Fig 1. Exposure of API from Cloud to be Consumed by Cloud/OP Apps

b. Exposure of Public API on On-Premise

Many on-premise enterprise solutions also publish APIs which can be consumed by various other satellite applications for extension and creation of partner application purposes.

There can be multiple ways of doing the same like:

•The data can be served in JSON format via a simple URL-based interface over HTTPS.

•using an API Gateway Architecture that can make the APIs available from publishers to the consumers.

There are many examples around us that provide open APIs in similar way like Currency Conversions APIs hosted on specific servers that can be consumed by any applications that create Apps

Figure 2 depicts how the Gateway provides a standard method for applications to expose data or custom logic as Rest based services in the on-premise system.

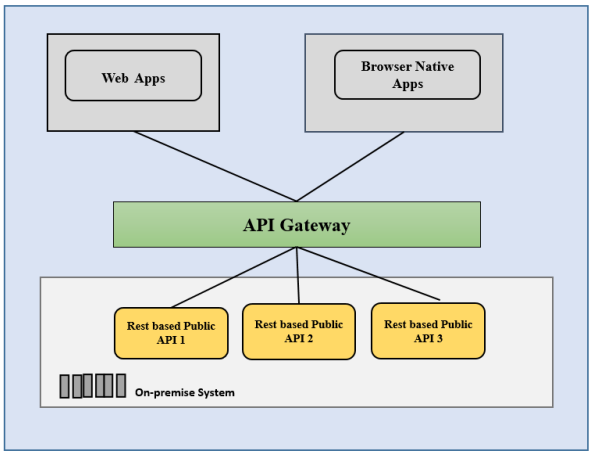

Fig 2. Exposure of API from On-premise to be Consumed by Cloud Apps

It maintains consistency in terms of security, monitoring and operations, versioning and documentation, and metadata exposure. Hence, the intent here is that every application must expose its core APIs without depending on additional components in the Cloud.

2.Private/Enterprise Cloud Services

In this segment, we discuss the scenario where there are applications hosted on private or enterprise cloud and need to communicate to applications/software behind the firewall in the on-premise world.

A. Reverse Proxy Mechanism

Reverse Proxy server basically requests resources on behalf of a client from one or many servers.

Many organizations create cloud connector which is a lightweight connection tunnel solution that can be used to connect applications hosted on private/enterprise cloud and existing on-premise systems.

Due to its reverse invoke support, the configuration is not required in the on-premise firewall to allow external access from the cloud to internal systems. The cloud connector runs inside a secure network and is placed on the on-premise system.

Implementation Design Example:

The reverse proxy mechanism is implemented via the Cloud Connector (CC) as depicted in Figure 3 which

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

connects between on-demand applications in the cloud platform and existing on-premise systems.
The reverse proxy enables CC to provide fine-grained control over:
•On-premise systems and resources (HTTP or RFC) that shall be accessible by cloud applications.
•Cloud applications that shall make use of the Cloud Connector.

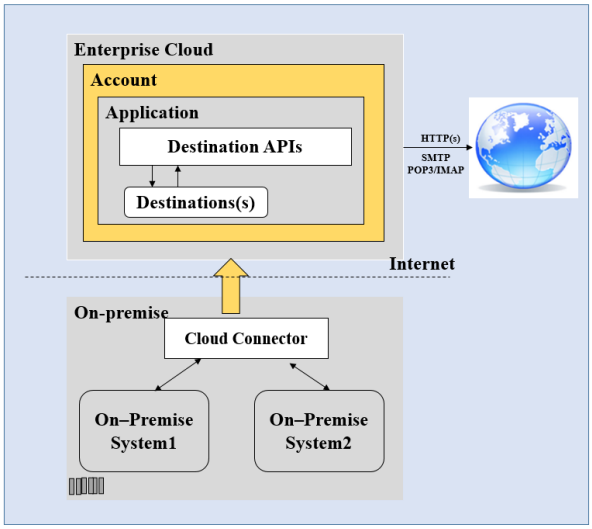

Fig 3. Reverse Proxy as an Enterprise Service Mechanism

This mechanism can be used by enterprises to connect on premise applications and cloud applications without opening the firewall.

B. Process Integration Services

The process integration service allows us to integrate various business processes spanning different companies or organizations. An event triggered from a business process in one application will call real-time or near real-time fetch of data from the follow-on business process in another application.

Implementation Design Example:

In the below implementation example, the Process Integration(PI) middleware is hosted on cloud. This is a general practice where PI services are hosted in an enterprise cloud like environment which help in bridging the communication between an on-premise(OP) system behind the firewall to a hosted cloud application.
As shown in the figure, there can be multiple processes executed between various systems and applications.

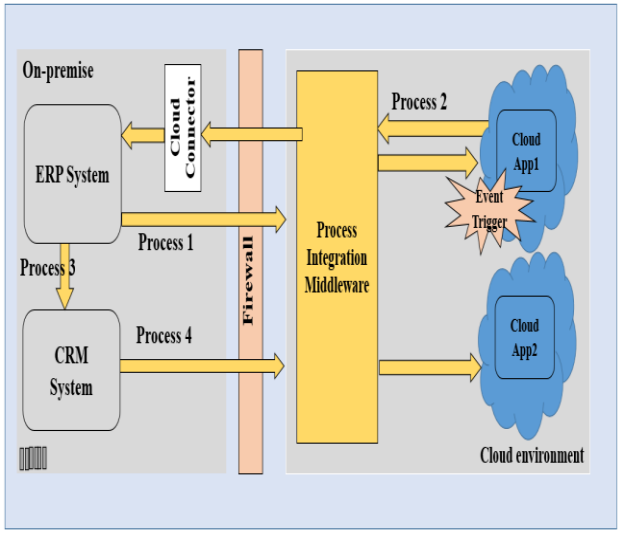

Fig 3. Process Integration via the Private Cloud

It follows one of the three patterns:

•Initiate and wait
•Notify and continue
•Synchronous request/response

Here, the data model, messaging patterns, and technical protocols are aligned between the involved applications. Process Integration can be used for a cloud to cloud or for a cloud with on-premise integration scenario.

C. Data Integration Services

The data integration service allows us to securely extract, transform and move data between on-premise and cloud systems. The objective of such a service is to allow efficient synchronization of bulk data using a rich set of transformation capabilities.

Implementation Design Example:

Data Hub Services are very prominent example of Data Integration. In a scenario where a software hosted on an on-premise system requires demographic information along with mash up of city, country, currencies, health statistics etc., processed information from heterogenous sources is required. In such cases Data Hubs hosted on Cloud are used so that many point-to-point connections between callers and data suppliers do not need to be made.

Apart from Data Hubs, there are many other ways to realize Data Integration Services, which typically also include a combination of creating a tunnel and consuming

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 1, January 2018

a service hosted on Cloud which gets data from various sources using a virtual schema mechanism.

D. Microservices

The microservice style of architecture which involves building applications as a set of autonomous services is tailor-made to meet the growing need for fast innovation in cloud computing.

Implementation Design Example:

The distributed microservices are built on in-memory data layers which offer memory-based and disk-based relational stores and real-time data transformation.

Figure 5 depicts the implementation of the Quantity Conversion microservice, which is an autonomous service hosted on the Cloud Foundry platform. This will efficiently enable industry cloud applications to perform quantity conversions at the required standards.

Data integration and real-time reporting on operational data are achieved by differentiating between the private data model of the microservices on one hand and data layer APIs on the other hand. Here, data model comprising of customer configurations is local to this microservice and is not shared with other services.

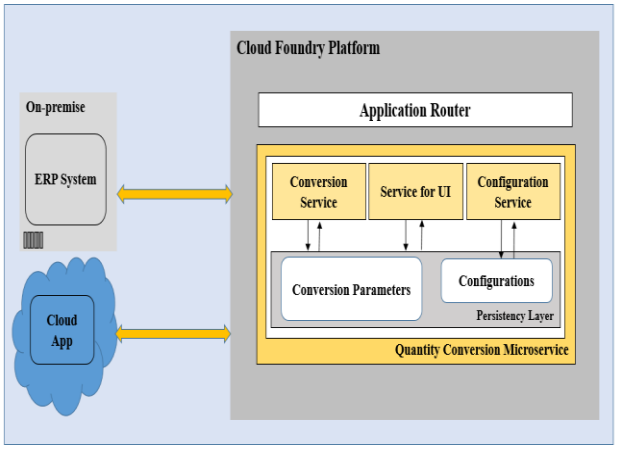

Fig 4. Microservice as a Private Cloud Service

The data layer API, comprising of the conversion parameters is restricted to read-only access and is published for use by other microservices. The conversion service built via the Java-based Spring Boot framework exports the configuration settings to the import service which in turn requests the third-party conversion routines to perform the standard conversions.

Similarly, microservices can be hosted completely on cloud and can be efficiently consumed by various enterprise solutions.

OUTCOMES/CONCLUSION

Different ways of integrating enterprise software solutions on Cloud have been implemented to help industries move towards shared computer processing to achieve coherence and economy of scale.

REFERENCES

[1] Sam Newman, "Building Microservices", O'Reilly, 2015.

[2] http://martinfowler.com/articles/microservices.html

[3] https://en.wikipedia.org/wiki/Cloud_computing

[4]https://en.wikipedia.org/wiki/Open_API

[5] https://en.wikipedia.org/wiki/Reverse_proxy