Refactoring Oracle Integration Cloud (PaaS)

Rajalakshmi Thiruthuraipondi Natarajan

rajalan11@gmail.com

Abstract

Oracle Integration Cloud (OIC) is a clous based solution provided to connect different applications, databases, and processes within cloud and with on-prem systems to securely and seamlessly connect and interface data and controls within the IT environment. It simplifies the connectivity between applications and automates the processes and workflows, thereby creating effective business flow. Using native connectivity APIs and in-built adapters, they can seamlessly connect to Oracle services and third-party services, as these provisions, eliminate the need for any complex coding and other configurations for establishing the connection. Also, with a comprehensive visual dashboard, a complete view of all the integrations is provided for monitoring and reporting the correctness and performance of the entire network. With evolving technological landscape and induction of new systems, new complexities might be introduced that might hinder the overall performance. Hence, here is a need to periodically review and refactor the integration services to simplify and optimize the solution by considering the overall architecture and the business needs. This provides a chance for IT team to deep dive into the integration solution and document any gaps that needs to be rectified and the integrator to adopt itself to the latest technologies and environment.

Keywords: Oracle Integration Cloud, OIC, Oracle PaaS, Oracle Middleware. Cloud Integration, Multi-environment interface, Applications Data Transfers, Oracle Cloud Suite, Oracle Cloud Development Tools, Oracle Cloud Refactoring.

Introduction

In reality, most organizations do not depend on one single solution for managing their entire operations and other activities. There are several applications, databases and other solutions wired to gather to cater to the custom need to the company. These diverse systems, constantly interact and exchange data amongst each other for performing their task and to support the business as a whole. The efficiency of the application and by extension the IT systems, depends largely on the ability to interact between the systems in a fast and secure manner. Oracle Integration Cloud is one of the leading cloud-based integration solution that is designed to seamlessly connect with diverse systems both within the network and to third-party systems. With its industry best APIs and adapters, developers can easily build and deploy integration solutions for their applications. With its unmatched reliability and stability, coupled with its extensibility makes it a formidable choice. However, with the adoption of new systems and technologies into the IT landscape, there is a need to revisit the OIC solutions to ensure that it is functioning at its optimum level and if not, redesign and refactor the solution, to improve the overall performance and other needs. There are several factors to consider while refactoring such as, simplicity, optimization, compliance, best practice, extensibility , and many more. By having a wholistic view of the operations and future scope, the integrations needs to be modified such that it does not break the existing functionality or the security but improves.

The Need for Refactoring

Through adopting a practice of continuously monitoring the IT landscape and refactoring it to perform its best, it is practically impossible and there are numerous factors that deter an organization from looking into refactoring or redesigning their applications. In fact, most organizations do not consider refactoring unless there are complaints with the existing systems or a new project it started for either dynamically changing the landscape or introduction of new system as part of the whole architecture drastically changing it. With the integrating systems being the nerve center of the whole IT landscape, it is imperative that these are refactored when the opportunity presents.

One of the main reasons for refactoring would be to optimize the integration's performance. With ever expanding it landscape the initial assumptions and the configurations that are put in place for an effective integration might no longer work at its best or be outright obsolete. Hence it is important with the addition of any new system or infrastructure or upgrade the integration points are investigated to check if they still perform the same way as intended. On the other side with new technologies the old standards or the expectations might not be relevant anymore. With the volume of data ever increasing and recent technology available to transmit large number of data seamlessly the existing limits and the SLAs need to be revisited to make sure that the integration is remediated to provide the optimal performance with the help of the latest technology.

Improving the integration’s readability and maintainability is another factor that might influence refactoring effort. With the gradual growth of the IT landscape new applications and infrastructures are added along with it the integration points extend to. Overtime these extensions become complex and uncontrolled that it becomes extremely hard for the support and the maintenance team to monitor or manage these systems. Any problems or issues faced with these systems becomes extremely hard to debug given the numerous points and the complexity it comes with. To counter this problem, it is important that the next integration systems are and revisited and simplified so that it can be easily maintained. Apart from maintenance simplifying the integration network would mean smaller hubs and the chance of data loss and latency can be greatly reduced. Error detection and resolution he said another reason that these systems need to be simplified since with complex systems the teams might be very hesitant to make any changes due to the fear of unknown.

The other relatively lesser reason for refactoring would be to adopt industry standard in defining the integration. The latest technologies such as Oracle integration cloud or any other niche technology are quite new to the market that it takes a while for the market to try and identify the best solution that can be adopted. It is quite possible that the implementation team might not have followed such standards and have deployed the solution that would merely satisfy the requirements but not in the best way possible. In this case the integration systems will be revisited to identify the deviations in the implementation and rectify to match the best practices defined by the market. While this practice is subject to debate since every organization is unique and their requirement might not fit into the template defined by the market such standardization effort would enable for better visibility and for support since this probably might be a language that every technical resource can comprehend. The ability to scale and extend the existing integration system is another key factor in the concentration for the remediation activities. If the system is not extensible this would mean a significant limitation in the prospect of the IT systems expanding. Hence any at all such limitations can be addressed as part of the refactoring effort to finely tune it and make it available for its effective operation in its current world and the changes that might come forth in the near future.

Refactoring Oracle Integration Cloud – Considerations and Best Practices

With the need for refactoring the integration identified the next logical step these two looks at the existing landscape and take the decision and the direction to move forward. There are several factors that needs to be considered before proceeding with the refactoring activity since this potentially might not produce any monetary benefit to the organization itself would rather simplify and ease the work of the IT team hence the team needs to decide and we in on different influencers to ensure that the benefit outweighs the cost. The first factor is the cost by itself for any project the organization needs to allocate a certain amount of budget and resources for completing the activity. The cost factor cannot just be looked in from just the integration point of view but rather broadened to identify the cost associated with the systems they connect and the maintenance costs.

The next factor is the technology by itself. It is not always the best path forward to adopt the latest technology. There are times that these technologies might have some serious bugs and might cause more problems than they solve. There is also a question about the compatibility. With the introduction of innovative solutions by either refactoring them to adopt new technologies or rewiring them, the ability of the other systems to effectively leverage these changes need to be considered. By rewiring the integration systems, the data flow is effectively disrupted which might lead to unforeseen consequences. Some of the in-transit activities that might not be properly documented or visible might be lost in the entire process. Since Oracle integration cloud links not just the system but workflows and the process additional caution needs to be taken to ensure that the process outcome are not disrupted as part of the whole refactoring process.

The turnaround time for the entire refactoring activity to complete from start to finish here is something that the project team needs to take into account. Since rewriting effectively means that the existing system is revisited to either tune or to adapt recent technologies and provide best benefit to the organization as a whole, it is important that the turnaround time be as short as possible. This does not mean that the quality or the requirements be compromised in any way but merely that given the target requirements and the expectation out of rewiring the system should be ready and functional as soon as possible. This project time should also consider the hyper-care Effort and the time that might be needed to stabilize the system. A longer it takes for the rewiring to complete it is quite possible that there could be new systems or a recent technology that might come in and make the whole project worthless.

No solution can be completely future proof. Every technology will become obsolete at a certain point and will have to be replaced by new ones. However, with careful planning and vision the systems can be designed in such a way that they are easily extensible and scalable to adopt new technologies and incorporate seamlessly within their network. And refactoring effort should do just that. The refactoring process should not merely be simplifying or reshaping the existing structure but rather ensure that the rules are defined in such a way that these can cater to the current needs of the organization while being prepared for the future. There are various criteria for future proof such as new technologies coupon hardware kappa availability of resources to maintain and support these recent changes come on ability to perform further rewiring, etcetera.

The above pointers are merely a few of the many considerations that the organization needs to take before moving ahead with refactoring their integrations. One needs to think from every angle of the technical architecture within and outside the organization that they talked to take conditions of rewiring. That could be a simple matter of timing that could throw the entire system of since the data either arrived too early or late because of this refactoring effort subsequently making the downstream system susceptible to failures.

The best practices are general guidelines provided by various implementers who based on their experience in performing certain task of the problem that everyone might face and the potential solution to avoid the same. While these guidelines can be used for having a sense of direction one needs to remember that every organization is different and are wired differently and there will always be a certain amount of deviation from the suggested solutions that they will have to adopt to best fit their organization. However, following or some of the industry accepted practices and thoughts that each implementer should keep in mind while rewiring –
Oracle integration cloud uses adapters agents, inbuilt APIs to connect with various applications and databases. These adapters and agents enable them to connect to external systems and other applications that are implemented within the Oracle cloud. The rewiring activity should always look for the latest APIs and the adapters available for usage period since oracle designs their solutions for the best performance it is a best practice to adopt and leverage Oracle provided solutions into the IT landscape. There are chances that due to

the unavailability of API or adapters come on there could be custom solution put in place for establishing this connectivity and transmitting data. However, since Oracle is continuously striving to provide the latest solution for its customers the number of adapters the agents and the APIs that they release are expanding to accommodate more touch points and third-party systems which the team can use to communicate with various systems. It is also important that as part of the API release there are certain functions that are newly added and some depreciated for several reasons. These APIs have to be identified and then replaced so that the system is at par with the latest release.

The next best practice is to always consider the predecessors and the successors in all aspects ranging from compatibility to the type of data provided the service provided and the time dependency. It could very well be that the delay in rendering the data to the successor system might be intentional that they expect the data to be available in a certain time and providing it earlier might not be in the best interest as a whole. The compatibility of the source of the destination systems should also be considered since eliminating the hops and directly integrating these two systems might mean that there are certain conversions that happen along the way might be bypassed thereby making the data unreadable or invaluable to the target systems. These problems arise typically due to the lack of understanding of the entire flow. Hence it is always best that the integration maintains and performs the most generic activities that can be returned for any systems and the target specific conversions and activities are retained within the application itself. Performing such complex calculations or translations within the integration hub might lead to the inability of refactoring because of the absolute need of those integration points or making it not scalable or extensible since a lot of the target calculations are performed within this integration points. This would defeat the whole scalability and extensibility feature of Oracle integration cloud and potentially leading to the creation of an extremely complex integration environment.

Another technical suggestion would be that every integration designed irrespective of OIC or anything else come on these two define a proper request and response design. That is every connection on the data transfer that happens receive the acknowledgment back confirming the transmission was successful. On the contrary come on there are asynchronous transactions or batch processing which follows the send and forget methodology where the data is sent but no notification is received back affirming the same. In fact, always he has a limitation that it does not support such asynchronous integrations. This brings about had not you need banned these responses should be received within a stipulated time. For instance, that response is not received within a time period of 5 minutes OIC assumes that the integration has failed, forcing the designers to build a system that does not run for an exceedingly long time. This needs to be particularly considered while transmitting large volume of data among various systems that might take longer than this duration. To counter this problem, it is always advisable that the data B cut into smaller chunks and sent to the respective target system and response received confirming the same.

Conclusion

As discussed above, refactoring is a continuous activity applicable of all sections of an organization’s IT world. When rightly done, this would ensure the integrations running smoothly with no issues. Times have changed, where the type of data, latency minimization, and the accuracy expected out of these integrations are higher than before and with terabytes of data passing through each day or even hour, the integrators needs to be working at its best to handle such volume, safely and quickly, hence justifying the need for continuous refactoring. However, it does come with a cost, which is an expense to the company. There needs to be a significant budget, and dedicated resources allocated to perform such refactoring activity, not to mention the time it take for the new system to stabilize. The is also an inherent risk of adopting a technology that might be too new that it might end up being counterproductive. Hence, a careful planning and analysis is necessary before initiating such refactoring activities. Predominantly, these refactoring is a manual activity, driven by external factors or change in the architecture. With the recent growth in AI and related engineering, the is a growing interest in using these technologies to measure and guide such redesigning activities. Though these technologies are at its infancy, it is quite evident that there are here to stay and with new innovations and methods to perfect them, AI and ML can soon be a valuable partner in refactoring.

References

1. Oracle Integration Cloud Service on Oracle Cloud Machine, https://docs.oracle.com/cloud-machine/latest/intcs_gs/index.html
2. Introduction to OIC (Oracle Integration Cloud) | What is OIC ? | Oracle integration cloud, https://www.youtube.com/watch?v=8YNl2zZZHys
3. Atul Kumar, Introduction to OIC (Oracle Integration Cloud) | What is OIC ? | Oracle integration cloud, https://k21academy.com/oracle-integration-cloud-service/introduction-oracle-integration-cloud-service-oic/
4. Application Integration, https://www.oracle.com/integration/application-integration/
5. Design Best Practices, Using Integrations in Oracle Integration 3, https://docs.oracle.com/en/cloud/paas/application-integration/integrations-user/common-integration-style-pitfalls-and-design-best-practices.html
6. The Definitive Guide to Oracle Integration Cloud Best Practices, https://appstekcorp.com/blog/the-definitive-guide-to-oracle-integration-cloud-best-practices/, Nov-2023
7. Oracle Integration Cloud: Features, Challenges, Benefits, and Implementing Best Practices for Success, https://www.korcomptenz.com/blog/oracle-integration-features-challenges-benefits-best-practices/#:~:text=Benefits%20of%20OIC&text=OIC%20is%20a%20potent%20solution,management%2C%20and%20enhance%20overall%20productivity.
8. Ishan Agarwal, Oracle Integration Cloud Gen3 Projects: Features, Limitations, and Advantages, https://appsassociates.com/blog/oracle-integration-cloud-gen3-projects-features-limitations-and-advantages/#:~:text=Although%20OIC%20supports%20a%20wide,party%20integrations%20or%20custom%20connectors.&text=While%20OIC%20supports%20version%20control,require%20careful%20planning%20and%20execution.