



7 Key Features in an Enterprise-Class Integration Solution

Contents

• Introduction	1
• Features of a good integration solution: Reliability, Consistency, and Scalability	2
1 Architecture	2
2 Consistency Model	3
3 Reliability and recovery mechanism	3
4 Ability to reconcile data	4
5 Failure and Conflict management	4
6 History sync support	4
7 Security and Audit support	5
• Conclusion	5

Introduction

Customers, today, want quality product, faster delivery, and seamless experience throughout their journey; and therefore, organizations are doing everything possible to accelerate their time-to-market without compromising on their product quality. The first step towards this endeavour is the creation of a unified ecosystem by bringing all tools and teams together – and getting rid of time-consuming manual collaboration process which leads to delays and sub-optimal decisions because teams do not have the full context. If you are reading this white paper, most likely, you are already aware of the struggles of operating in a disconnected ecosystem and looking for ways to integrate the heterogenous elements in your ecosystem. It is also possible that you, already, have an integration setup that is unable to help you achieve the desired level of collaboration or productivity, and you are trying to figure out what's gone wrong with your choice of the integration solution.

As integration is a relatively new space and organizations have only recently started to realize its importance, there is a great deal of confusion about what to expect from a good enterprise-class integration solution. In this whitepaper, we will discuss the features of a good [integration solution](#) in detail.

Features of a Good Integration Solution: Reliability, Consistency, and Scalability

An integration solution plays a huge role in keeping all the [tools](#) and teams in the ecosystem together – but it runs in the background, has no interaction with people and shouldn't require any change in the end user behaviour. Therefore, the [integration solution](#) should be reliable and consistent and run successfully without much supervision or monitoring.

Another quality that is a must in a good integration solution is its ability to scale for future requirements. While your current requirement might be to establish a bidirectional integration between Azure DevOps Server (TFS) and Jira to sync defects from one system to another in real time, in future, with more and more tools getting added to your ecosystem, you might need the integration solution to integrate thousands of projects and entities from hundreds of other end systems.

To determine whether the integration solution is reliable, consistent, and scalable, you should evaluate the solution on the following parameters:

1 | Architecture

The architecture of the solution should be such that it supports scalability and data consistency: The architecture must support integration of the solution with variety of end systems, such as Application Lifecycle Management (ALM), Customer Relationship Management (CRM), Source Code Management (SCM), IT Service Management (ITSM), and so on.

- It should allow the solution to support vast variety of entities and customize the sync flow, if required.
- It should support configuration, admin, and run time performance.
- It should be easy to support future changes in the integrated systems.

2

Consistency Model

The consistency model on which the integration solution is designed determines the kind of consistency it can provide while synchronizing the data between multiple systems. Some common consistency models are:

- Always Consistent model: This model ensures that the data between two systems is always in consistent state.
- Eventual Consistency model: This model ensures that the data between two systems will eventually be in a consistent state without manual interference.
- Best Effort Consistency model: This model ensures that the integration server will do its best to keep data between two systems consistent, but can't guarantee consistency in case of failures.

3

Reliability and Recovery Mechanism

One of the fundamental challenges with integration solution is that the reliability and recovery mechanism needs to be built-in every aspect of the solution. The reliability and recovery mechanism in a good integration solution must ensure that:

- In case a change is not completed due to connection loss, it should be completed without any manual intervention when the connection is restored.
- In case of any failure during integration, the integration should be automatically retried without causing any data loss.
- If the integration server goes down during syncing an event and someone changes the same entity in target system at the same time, the integration server should be smart enough not to overwrite the changes.

4 | Ability to Reconcile Data

In a dynamic [ecosystem](#) where multiple concurrent changes are happening, any problems with data mapping or conflict resolution can lead to inconsistency between the systems involved. Therefore, the integration solution should have a defined method to detect out-of-sync entities and consistent rules to be applied for such entities. The reconciliation rules should be so robust that there is no scope of duplication or data loss even when the data is being moved from a manual integration platform to an [automatic integration](#) platform.

5 | Failure and Conflict Management

In case of any failure, the integration solution should allow the administrator to easily take corrective actions for all the entities in the failure queue. The administrator should easily group similar entities and take bulk actions to resolve it. It is also mandatory to provide configuration option for automatic retries of failures. The common conflict resolution rule of Master/Slave is not ideal for all kinds of data in the ecosystem. Therefore, the support for conflict detection and alternative ways for configurable conflict management is a must-have feature in a good integration solution.

6 | History Sync Support

Applying the individual change sets (and not just the current state) is critical to ensure that the history of the destination system reflects the full history of all the changes. Therefore, a good integration solution should be able to mirror the changes in the same grouping and order from the source system to the target system. Any inconsistency can, otherwise, lead to wrong reports, incorrect response time, cycle time reflected in the end systems.

7

Security and Audit Support

An enterprise class integration solution can't be considered reliable until it guarantees data security and secure user access. Therefore, the data/credentials stored within the integration solution should be in encrypted format and there should also be mechanisms in place to ensure secure user access. The administrator should have the ability to audit all kind of changes, control user access and assign different privileges to different users.

Conclusion

A good integration solution is the backbone of a highly productive and collaborative ecosystem. Therefore, you must evaluate an integration solution on all parameters to ensure that it is reliable, consistent, and scalable. While the choice between: going with an inbuilt integration provided with a solution, building an integration solution from scratch, and buying an out-of-box integration platform is yours, you must weigh the advantages and disadvantages of these integration strategies with respect to your requirements and the feature completeness of the solution. Two key aspects that you must consider while making this choice is the level of productivity and collaboration that the chosen integration strategy helps you achieve and the investment in maintaining a high-level of productivity and engagement in future using the same solution.

[OpsHub Integration Manager](#) is the flagship integration solution from OpsHub that has all the 7 features discussed in this article. OpsHub Integration Manager improves the efficiency and effectiveness of agile teams in ALM and DevOps environments by making necessary and current data available to each user, in that user's preferred system, with full context, in real-time. This, in turn, speeds up the development processes, reduces errors, improves decision-making, and results in delivering innovative products and services faster, with higher quality at lower cost.

To know more about OpsHub Integration Manager (OIM), schedule a [complimentary consultation](#) session with our integration experts.