



# **Getting Started** with Integration Architecture

Also known as Enterprise Application Integration, Integration Architecture provides a method for different applications to communicate and share data. This breaks down data siloes and provides visibility over application landscapes and data flows.

The modern enterprise will, on average, use as many as 900 different applications in their day-to-day processes. For the employee who may have to switch between dozens of different applications, this has the potential to severely curtail productivity. In order to break down the barriers between software, organizations employ integration architecture, or enterprise application integration, to enable applications to connect and share data.

Integration Architectures have been around for many years, with many organizations using an ESB, or Enterprise Service Bus, to provide communication. However, as the complexity of technology has grown and new development styles have arisen, the ESB model has fallen out of favor, unable to keep up with the pace of change. Modern enterprises are increasingly turning to integration platforms as a service (iPaaS) or utilizing a microservices approach and attempting what is known as agile integration.



## What is Integration Architecture?

An Integration Architecture is the set of applications and technologies that bind together disparate applications, enabling them to communicate and work together. For a very simple example, consider Microsoft Office. Word and Excel are different applications with different file structures, but it is still possible to insert an Excel table into a Word document that is linked back to an existing spreadsheet, due to Microsoft's integrations. An integration in which two applications are directly linked is known as a point-to-point link.

Of course, actual Integration Architecture involves much more complexity. An Enterprise Service Bus needs to deal with hundreds of applications, using different programming languages, APIs, often having to integrate legacy technologies, all in real time. Indeed, one of the major reasons that the ESB model has fallen out of favor is that ESBs could not keep up with the development of modern software, and shifts to cloud applications and agile development have led to iPaaS and Agile Integration. An iPaaS is a cloud-based integration solution, allowing for on-premise and cloud applications to be integrated. Typically, both an ESB and an iPaaS solution will be middleware, acquired from a third party firm. It is not infeasible for an organization to develop their own integration system, though it can be very resource heavy. Firms that do proceed down this route are increasingly turning to agile integration, an approach built around the agile development philosophy, often utilizing microservices architectures.



## What are the benefits of Integration Architecture?

This is a fairly simple question to answer – enabling applications to communicate and share work significantly smooths the work processes of employees. Having to consistently deal with data problems and maintenance would be taxing even for experienced engineers, let alone employees without computer science backgrounds (in fact, one of the advantages offered by many iPaaS systems is that they are low code environments, meaning even laymen can create or adjust integrations). The architecture itself will still need management, whether from architects or developers, but this is significantly lower footprint.

Not only that, but many integrations enable automation of tasks to further improve productivity. Consider a standard task: following up with a new sales lead. Integrations are what enable a website or CMS to communicate with your CRM and record the lead's details. Further integrations can enable the CRM to automatically notify a sales member through email or a chat app like Teams, while setting a reminder in the calendar for a second follow-up. These kinds of things can be small, but may save a lot of time over a large scale.





A related benefit is the possibility of maintaining support for legacy applications that may still have important uses for the firm. Old and outdated applications can be important to certain divisions for a wide number of reasons, and enabling those applications to survive for longer periods can save the firm from difficult decisions, and enable them to spread out the rationalization of applications.

A powerful integration platform can give the IT department more flexibility when it comes to choosing applications. A lot of modern software companies attempt to “bundle” customers, signing them up to a variety of related applications because they can offer easy integration. In the above example, you could avoid having to do any integration through by choosing a supplier that offers both a CMS and a CRM. This can lead firms to be almost forced into certain configurations, but with a good integration solution it becomes a lot easier to pick and choose and avoid having to be vertically integrated.



## The Challenges of managing Integration Architecture

We've spoken above the costs and development resources required to build an Integration Architecture for an enterprise, but as iServer is an enterprise architecture tool, this section will focus specifically on the challenges faced by architects, rather than developers. Perhaps the biggest issue for architects is actually viewing and understanding the architectural landscape. Given the size of enterprises, the requirements to understand the applications, technologies and systems could be documented across a vast array of different elements and people. A similar issue arises from the tracking of data flows. Poor and inconsistent documentation of data flows leads to impossible data lineage, and prevents any kind of impact analysis upon the application landscape.

One overlooked problem is the mapping of architecture efforts to business strategy elements. The end goal of integration architecture should be something that fits well with an organization's people and processes, not merely the best technical solution. However, firms can be guilty of neglecting these elements, which can then impede on the application architecture, as well as the alignment of data flows with business strategy. Given the pace of change in the technology sphere, it should come as no surprise that integration architecture efforts also need to be constantly updated to remain useful. Most Enterprise Architecture teams already have a wide range of key tasks to manage, and this place a further burden on their time and ability to complete other initiatives.





## How does Integration Architecture interact with iServer?

Continuing the focus on enterprise architecture, how would the process of managing integration architecture look using an EA tool such as iServer?



## 1. Information Gathering

The first step is to establish a baseline architecture, by loading all relevant information into the central repository, which could include the likes of existing available Application communication diagrams, data flow diagrams, API catalogs, matrices and third-party tool exports. You'll also want to establish goals and parameters for the initiative.

What will the scope be for this exercise: Application Communication and Complexity? Interfaces Modeling? Data Lineage?





## 2. Assess the current landscape and determine where integrations are needed

Next, architects will need to assess the requirements of the integration architecture and work to implement them. Typically, architects will need to model any integrations according to a metamodel, which iServer provides. Further analysis may also be required, with ready-to-use templates for impact analysis and dependency visualizations of data flows, applications, interfaces and CRUD operations all provided.



### **3. Enable Business Users to view information and keep it up to date**

The purpose of integrations is to allow business users to navigate the application portfolio without having to be IT experts or demand IT resources. Naturally, keeping users in the loop is an important final step. iServer comes a variety of Integration Architectures views to display information, easily available to all stakeholders through the central repository. You can quickly create interactive and tailored navigation pathways for stakeholders to consume integration architecture deliverables.

## Maintaining an effective Integration Architecture

The application landscape in an enterprise is not static, which means any integration architecture will need to adapt to the needs of the business as it evolves. Of course, maintaining systems like this is one of the main purposes of an Enterprise Architecture department, whether for Integration Architecture or any other part of EA. Nonetheless, it's always a good plan to maximize efficiency for these areas. There are a number of steps architects can take to help prevent constant maintenance from being a resource drain.

The first is simply to outsource your integration efforts to an iPaaS provider. As a SaaS product, an iPaaS will possess an evergreen roadmap of updates and support. This does mean less flexibility and control, but is one of the principal reasons that organizations have opted for these types of programs. An advantage that many iPaaS applications possess is a low code or no code environment, which drastically reduces the barrier to entry for creating or improving integrations between applications. However, this is not a feature exclusive to iPaaS, and can be implemented in other Integration Architectures to reduce the strain on IT and EA when it comes to maintenance.

As mentioned above, another step is simply to keep business users informed and able to view and edit integration architecture documentation. By highlighting any issues or changes needed, a lot of legwork for architects can be spared.



## Summary

Integration Architecture is at once a complicated and simple area. On the one hand, it's merely a way to keep different applications linked and communicate data across the business. But on the other hand, building and maintaining an effective integration architecture is one of the most technical and time consuming parts of the Enterprise Architecture function. Hopefully this guide will help to clear up any initial confusion over the subject, and provide some handy tips for managing integration architecture in the future.



© Copyright 2020 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to: [marketing@orbussoftware.com](mailto:marketing@orbussoftware.com)

**Orbus Software**

Floor 4, 60 Buckingham Palace Road, SW1W 0RR

+44 (0) 20 3824 2907

[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)

[www.orbussoftware.com](http://www.orbussoftware.com)