

# Integrated Integrations: **ServiceNow**



# Introduction

There are dozens of applications which are vital to everyday productivity in organizations. However, no application is powerful and versatile enough to take care of every need in a business.

Employees will inevitably be working across several pieces of software to accomplish tasks, which inevitably presents a problem. How do you best transfer work, or data, between all these different applications? Increasingly, the answer is via integrations, whether built-in or custom made.

Microsoft 365 is a classic example, with each application in the suite integrating seamlessly with each other. There are no problems pasting a table from Excel into a PowerPoint spreadsheet, sharing the final deck on Teams and uploading it to SharePoint. Everything is smooth and simple. That's the kind of experience that everyone wants all the time.

Of course, it is easy for Microsoft because they control the entire suite end-to-end. Problems arise when you step beyond the "walled gardens" of the major software providers.

The ServiceNow logo, consisting of the word "servicenow" in a lowercase, sans-serif font, with a green dot over the 'o'.A small icon inside a white box, featuring a stylized 'i' and a bar chart, representing data or information.

**What happens if you need to move data from a CMDB like ServiceNow into an Enterprise Architecture tool like iServer365 in order to run analytics in a Business Intelligence tool like PowerBI?**

It's here that the hard work begins.

# Leveraging Every Tool

The above example is actually one of the more simple situations when it comes to integrating different tools. Why? Because there are built-in integrations at every stage. ServiceNow has a series of integrations with iServer365.

The ServiceNow CMDB stores up-to-date information about IT assets such as business applications, software models and hardware models. iServer365 keeps its repository updated by importing this data on a scheduled basis. Automating this process eliminates the chances of human error and ensures that the data held in iServer365's repository is accurate.

The ServiceNow logo, featuring the word "now" in a lowercase, sans-serif font with a green dot over the 'o'.

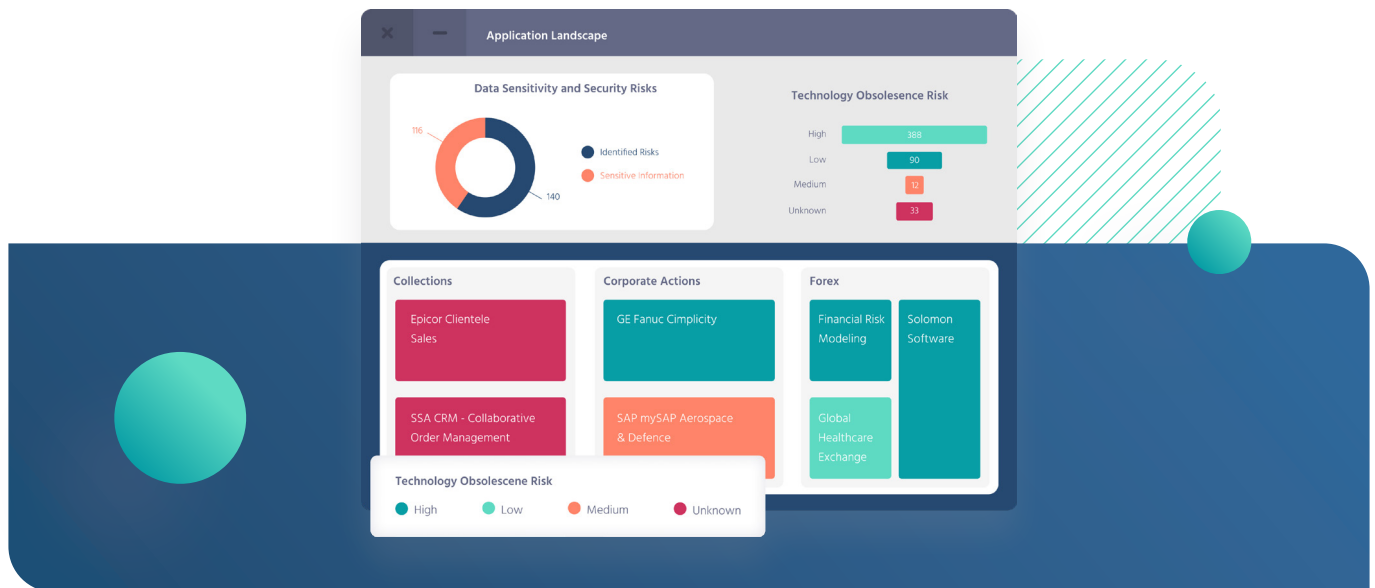
Meanwhile, iServer365 has been built from the ground up to work with the Microsoft 365 productivity suite. Both Microsoft and ServiceNow (and iServer365, for that matter) offer ways for other software providers to integrate with their platforms, with a variety of partner programs in the technology sphere. At the time of writing, ServiceNow's Partner Finder for technology integration lists more than 200 partners, while there are surely untold numbers of Microsoft technology partners – **one estimate** suggests there are as many as 90'000 partner companies in the Microsoft ecosystem, though not all would be technology partners.

However, for the smaller companies or less widely used applications, users will often have to turn to Application-Program Interfaces (APIs) in order to produce results. Across an enterprise, architects will need to develop integration architectures. 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 the purpose of Integration?

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. 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.

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.

One of the major advantages is enabling each and every application to play to their individual strengths, instead of forcing unfamiliar roles upon flexible applications. ServiceNow, for example, can fulfil a whole host of different tasks, but that doesn't mean it is designed to do so. One of the reasons for the built-in integration between iServer365 and ServiceNow is that each tool can operate as they were intended to. iServer365 is a high level digital transformation tool that provides oversight and control of initiatives. ServiceNow ensures that everything runs smoothly at a lower level, providing up-to-date information and a complete picture of the enterprise. By integrating the two, you can get the best result possible.

# What happens without Integration?

Let's assume that we don't try to do everything with just one tool. An enterprise gets ServiceNow to handle their IT service management and iServer365 to take care of architecture. However, it fails to properly integrate the two applications. What will happen?

- Decisions made on digital transformation are likely to be flawed as they are not based on the most accurate and up-to-date information.
- Architects spend a significant amount of their time trying to keep all of the data in the repository up to date but in reality can't because of constant changes to the architecture – many of which they may not be aware of.
- What could the impact of a bad decision be?
  - Loss of productivity due to business applications being unavailable
  - Loss of reputation if customer-facing services are impacted
  - Removing of an application believing that it is not used when in fact it forms a critical part of a business process.

The screenshot shows the ServiceNow 'Import Settings' and 'Field Mappings' configuration. The 'Import Settings' section has two dropdown menus: 'ServiceNow Source Table' set to 'Business Application' and 'iServer365 Object Type' set to 'Application'. The 'Field Mappings' section shows two rows of mappings. Row 1 maps 'Status' to 'Lifecycle Status' via an 'Attribute' mapping type. Row 2 maps 'Category' to 'Application Category' via a 'Relationship' mapping type.

	ServiceNow Field	Mapping Type	iServer365 Field
1	Status	Attribute	Lifecycle Status
2	Category	Relationship	Application Category

# Off The Shelf VS Custom Integrations

Making use of tools like ServiceNow that have hundreds of technology partners with ready made integrations is great, and part of the reason why some enterprise tools have remained so popular is their dedication to working with as many other tools as possible.

But does this mean you should only look for tools that have an easy collaborative ecosystem? Aside from possibly overvaluing the convenience factor, custom integrations may be an important capability when it comes to your own products. A key responsibility for many enterprise architects is maintaining the integration architecture that enables different platforms to talk with each other.

A mature integration architecture will be able to quickly adapt to new product requirements or changes to the application portfolio. Many enterprises make use of iPaaS, or Integration Platforms as a Service. A good iPaaS solution offers the benefits of traditional integration architecture, removing data siloes and enabling easier communication between applications. At the same time, an iPaaS solution can deliver faster time to value in comparison to other integrations, and have thus far offered lower costs to implement. Where iPaaS struggles mostly comes from the immaturity of the software; the likes of Microsoft's Azure Integration Services or Oracle's Integration Cloud Service have only been available since 2015, while many of the other vendors are still small firms with limited global coverage and support.

A third party iPaaS offers plenty of benefits, but may not be the solution for every enterprise. Internal development of cloud-based integration remains possible, but comes with additional considerations. First and foremost: are the APIs available to you? While every cloud application worth its salt will give developers access to an API, not every API will give you everything you need to comfortably communicate between programs – and even where they do, the amount of development work necessary may be prohibitive. Or, documentation may not be adequate, even if the functionality exists.

Of course, in an ideal world you will be able to work with applications that offer a variety of ready-made integrations while still having powerful APIs available for individual integration needs. It is likely that these product features will become increasingly important for enterprise software as the importance of integration grows.



## Summary

Much of what we have said here will not be news to enterprise architects or IT professionals. Building integrations between applications through something like an Enterprise Service Bus has been an important role for decades, while it is obvious from any product website that firms value out-of-the-box integrations. However, those away from the core activities of IT may well be used to everything just working together, unaware of the massive amount of work that goes into enabling this.

This is where the enterprise architect and iServer365 can work in tandem to provide the organization significant value. iServer365 can be used as a central source of truth enabling engineers to document their application integrations at a higher velocity and prevent disparate data flooding the organization – through iServer365 the organization will have an increased oversight and modelling capability that will enable the organization to undertake integration changes at a higher velocity.



## See iServer365 and ServiceNow in Action

Book a Demo with iServer365 to see how easy it is to integrate  
your ServiceNow workflows with the market leading digital  
transformation platform

[Book a Demo](#)

### About Orbus Software

Orbus Software is a global software vendor and a recognized leading provider of cloud solutions for digital transformation. Its products drive alignment between strategy and execution by leveraging familiar Microsoft tools to ensure rapid adoption and best-in-breed functionality. Orbus Software's market leading iServer Suite provides customers with a strategic decision-making platform addressing key digital transformation disciplines, including Enterprise Architecture (EA), Strategic Portfolio Management (SPM), Business Process Analysis (BPA) and Governance, Risk and Compliance (GRC).





**+44 (0) 20 3824 2907**  
**[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)**

Floor 4, 60 Buckingham Palace Road, SW1W 0RR

© Copyright 2022 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)**