Streamlining business processes with SAP connectors and Azure services

Today, enterprises recognize the importance of adopting the cloud as part of their digital transformation. But to fully embrace this transformation, the cloud can’t be solely relegated to lower-value infrastructure services such as storage or backups; it must become a core aspect of every business process where it can leverage the cloud’s platform as a service (PaaS) and software as a service (SaaS) capabilities.

As a technical decision maker, what are you doing to extract maximum value from your cloud services? For example, do you have a Microsoft Azure cloud subscription and are any of your business-critical processes running in SAP? Do you have other business apps in Azure that would benefit from accessing data that resides in your SAP environment? How are your Azure apps accessing your SAP-based information, and what are the benefits and challenges associated with integrating these two environments?

This is what we’re doing at Microsoft: incrementally moving all our IT and operational infrastructure—including our business-critical SAP systems—onto Azure. This case study highlights the SAP Web Services Platform and the SAP Advanced Business Application Programming (ABAP) software development kit (SDK) for Azure that we developed to integrate SAP and Azure natively. These new connectors have greatly streamlined our development efforts to integrate SAP and Azure and have made our business-critical processes more manageable. Other organizations might consider leveraging this SDK to natively connect their SAP environment to their Azure cloud or following similar steps to develop their own SAP Web Services Platform for Azure.

Drive to integrate SAP environment with Azure

At Microsoft, as we continue our digital-transformation journey, we’re exploring a variety of ways to create new business value through the intelligent cloud and intelligent edge. Fundamental to this objective is a Microsoft Core Services Engineering and Operations’ (CSEO) mandate to move the entire organization to Azure. Recent milestones we’ve achieved as part of this effort include migrating our entire SAP environment to the cloud, driving digital transformation with modern network infrastructure, and transitioning our corporate network to be internet- and wireless-first.

Migrating all our SAP infrastructure to Azure was a monumental leap forward to becoming all in the cloud, but we also wanted to achieve a tighter integration between Azure and SAP. Doing so would accelerate the digital transformation across the business by enabling internal and external partners to connect their applications to our business processes more quickly and easily than ever before. We needed to develop connectors and APIs between SAP and Azure to:

- Simplify and modernize our IT landscape while promoting agile business processes.
- Streamline multiple point-to-point solutions into a single end-to-end architecture solution that would enable both internal and external partners on their digital transformation journey.
- Simplify Azure developers’ efforts to access SAP and SAP developers’ efforts to integrate with Azure.
- Offer real-time data availability on Azure services, and real-time access to SAP functionality from Azure applications.
Our platform and connectors integrate SAP and Azure

We approached integrating our SAP environment with Azure in two different ways:

- Building an SAP Web Services Platform for Azure that enables partner applications to integrate with SAP in a consistent and efficient way.
- Creating an SAP ABAP SDK for Azure that our SAP ABAP developers can use to natively integrate SAP systems with Azure data, tools, and services.

Here, we discuss these two different interfaces and how they are designed to accommodate two distinct groups of developers: the web services platform is for Azure developers, and the SDK is for SAP developers. Other organizations can also utilize the Microsoft-developed SDK to connect their SAP environment to Azure or build a similar web services platform with APIs for their own environments.

SAP Web Services Platform for Azure: Reaching into SAP from Azure

Challenges with the old system

In our old environment, internal and external partner applications that needed to access SAP from Azure had to work with a variety of different point-to-point solutions. As illustrated in Figure 1 below, this architecture was based on three different patterns to support internal, extranet, and cloud-based partner apps, and typically required middleware that involved multiple network hops to reach the data.

![Diagram](image)

*Figure 1. In our old environment, partner apps that needed to access SAP data had to work with one or more of three different patterns for internal, extranet, and cloud-based locations—resulting in a complex solution that necessitated multiple point-to-point hops.*
This complex environment was resource intensive because we had to support all three patterns at once. Infrastructure costs were also high, due to duplicated services required to run internal and on-premises applications vs. those hosted in the extranet. And with its dependency on middleware, security and the amount of labor required to maintain all the components were ongoing concerns.

This complicated mix of point-to-point solutions had evolved over several years, and we wanted to replace it with a completely new platform that would enable apps to interoperate between Azure and SAP in a consistent manner. This platform would be an end-to-end solution for our partner apps that would reduce resource costs and simplify the environment by converging three separate patterns into one. The solution also needed to be able to access Azure tools and services, such as Application Insights, API Management, Traffic Manager, and others to enhance operations.

What we built
To accommodate these design requirements and support our digital transformation through the intelligent cloud and intelligent edge, we built our next-generation SAP Web Services Platform for Azure using the SAP .NET Connector and deployed it to Azure as a RESTful service. As Figure 2 illustrates, this new solution is based on the enterprise hybrid connectivity (EHC) basic design pattern, utilizing Azure AD (Active Directory) and Kerberos authentication to enable access to the designated SAP system. This new interface provides a single means for partner apps to integrate with SAP no matter where they reside—in Azure, an extranet, or on-premises.

Benefits of using the new platform
Our new SAP Web Services Platform for Azure enables Azure developers to work with a single set of services to access all our SAP systems, including Enterprise Resource Planning (ERP), Master Data Governance (MDG), and Supply Chain Management (SCM) in real time. This platform reduces our infrastructure complexity and associated costs. It also has lowered our maintenance and support costs, because we’re no longer supporting multiple point-to-point solutions.

By leveraging the power of Azure, this platform provides all the capabilities that enterprises demand of business-critical systems: high availability, efficiency, and disaster recovery. As Figure 2 illustrates, the platform lets us take
advantage of the entire stack of native Azure tools—something our previous on-premises-based system couldn’t do. Through the platform, we can now use:

- **Azure Traffic Manager** to support geo-replication by routing traffic based on geographic location.
- **Azure API Management** to manage APIs in a consistent and efficient way. It not only enables us to maintain a better inventory of the APIs, but also helps us control access to the API by subscription.
- **Azure Key Vault** to store and manage highly sensitive data, such as passwords, connection strings, and other data.
- **Azure Application Insights** to obtain near real-time details on overall system health. This tool facilitates our telemetry and monitoring efforts by providing out-of-the-box logging and auditing. It also enables operations to be more proactive in identifying potential issues as early as possible—thereby decreasing our mean time to resolve (MTTR) issues.

**SAP ABAP SDK for Azure: Connecting SAP to Azure**

As part of migrating our business-critical processes to the cloud, our SAP development team needed to send data from the relevant SAP systems to Azure services, but had no efficient way to do this. At the time of this effort, no native integration mechanism existed. Although Azure supported numerous SDKs and tools for programming languages such as Java, .NET, Node JS, PHP, and Python, no SDK was available for ABAP, SAP’s proprietary programming language.

**Challenges with the old system**

Without an ABAP-specific SDK, our SAP developers had no out-of-the-box support from SAP to consume Azure services. In addition, aside from SQL Server Storage and Azure Blob Storage, there were no readily available products that could integrate SAP to Azure cloud services. In the absence of native integration, we had to incorporate middleware solutions to shuttle data between the two environments. Our developers would then engineer their own point-to-point solutions that entailed multiple network hops through the middleware—along with a lot of custom coding for:

- **Authentication.** As a cloud service, Azure employs sophisticated authentication and authorization procedures that are challenging to code correctly.
- **Data formatting.** Data formats are completely different between Azure and SAP—and also differ among systems and services. For each data transmission from SAP to Azure, our developers had to custom code the correct translation between the two messaging formats, writing custom code for Azure EventHub and different custom code for Azure Key Vault.
- **Consuming the end points.** Azure has REST endpoints to which data should be sent. But programming this can be extremely challenging, especially for ABAP developers who had to work in an unfamiliar environment where properly coding for a single REST endpoint might require more than 1,000 lines of code.
All this custom work resulted in a more complex and costly environment due to the extra coding, licensing, and maintenance requirements. Our SAP developers needed the ability to connect our SAP systems to Azure natively by using SAP ABAP language that would abstract the complexity entailed in accessing Azure services.

**What we built**

CSEO recently released the [SAP ABAP SDK for Azure](https://www.microsoft.com/itshowcase) to simplify SAP integration with Azure. Because this SDK uses SAP’s proprietary ABAP language, SAP ABAP developers can now use a familiar programming language to integrate their SAP systems with the latest Azure tools, services, and data directly.
Figure 4. Our new ABAP SDK for Azure enables our SAP developers to use a familiar programming language (ABAP) to easily and securely access Azure tools, services, and data.

Benefits of working with the new SDK

The new ABAP SDK for Azure simplifies and standardizes Azure access for SAP developers by abstracting the complexity that would otherwise be involved in coding for Azure. The SDK also follows industry standards to ensure the safety of secret keys that are provided by Azure cloud services. It also facilitates appropriate implementation of Azure authentication, authorization, shared access tokens, and other security routines.

With the ABAP SDK for Azure, our SAP developers can now:

- **Speed development time** by using their familiarity with ABAP to integrate their SAP environment to Azure.
- **Improve system performance**, because the SDK supports pushing near real-time data to Azure services.
- **Enhance system security** by ensuring that SAP functionality and data are available in a secure Azure cloud environment for partners and for other custom applications.
- **Help the business accelerate our Azure cloud adoption** by leveraging Azure more effectively for SAP-based business-critical processes.

Lessons learned and best practices

The journey that we undertook to create a world-class solution to address a complex set of problems that Microsoft and surely other enterprises face today was a challenging one. We share here the lessons that we learned and the proven practices that other companies might adopt in their organization’s efforts to connect SAP with their Azure environment.
- **Determine which interface you need.** What would serve your ecosystem best—crafting your organization’s own API? Working with the SAP ABAP SDK for Azure? Or both? It depends on your business needs:
  - **Develop a web services platform to enable Azure-based apps to communicate with SAP.** If you have various Azure .NET applications in your organization that need to integrate in near real time with different business processes in your SAP environment, consider developing an SAP web services platform to build APIs on top of those SAP business processes. Not only can your platform enable your Azure app to integrate with SAP, you’ll also be able to leverage native Azure tools and services that can greatly enhance management of business processes.
  - **Leverage the SDK to connect SAP business processes to Azure.** In a scenario where you have an SAP business process that needs to access Azure features such as Key Vault or to publish SAP events such as payment status to Azure EventHub, your SAP ABAP developers can leverage the new SDK to connect to these features quickly and securely.
  - **Consider the locations of your data when building out your solution.** Large enterprises commonly operate in multiple geographic locations. When planning your integration, be sure to review where your system(s) will run and consequently, which region(s) you should select when finalizing your architectural design. As an example, if your back-end SAP system resides in one Azure region, plan to deploy your Web Services Platform for Azure to the same region (or at least to a nearby region, if necessary).
  - **Assemble the right development team.** By its very nature, developing a web platform for Azure to reach into SAP, or working with the SAP SDK to enable integration with Azure, requires experts from both sides. We recommend engaging the following resources:
    - **SAP configurator.** This person defines and designs the entire business process and is the primary liaison with the business when determining business requirements for the solution. Your SAP configurator can also help your API developers determine the best initial targets (which environments and/or business processes) to develop, as identify what SAP data in which environment they should access and leverage when testing.
    - **SAP ABAP developer.** This person works with the SAP configurator to finalize the business process, business validation, and logic.
    - **SAP security team.** When the business executes a process through the platform to SAP, it makes an SAP connection. This team facilitates the required authentication and authorization of users against that business process.
    - **SAP Basis.** SAP administrators work with Basis to manage infrastructure, performance, and health of the overall service.
    - **.NET software Azure engineer.** This engineer is an expert with the .NET part of Azure and wraps the SAP business process in a web API.
    - **Networking/IT team.** This team helps resolve any networking issues that might arise when attempting to establish connections between SAP and Azure.
  - **Compare results from the new interfaces with previous results.** If you are replacing an older on-premises web service to SAP with a new API developed using this web services platform, you can vet the accuracy and performance of the new API by comparing its functionality to known results from your older solution. For example, when you call your old service, what specific result would be returned and in what timeframe? Capture these data and compare them to the results from your new service. Do you get the same results in the same timeframe, or in a reduced timeframe?

**Conclusions**

Microsoft Core Services Engineering and Operations’ migration of our entire SAP environment to Azure constituted a major milestone as the largest and most complex workload migrated to Azure to date. In our ongoing journey to transform the company’s operations to support the new digital world of business, we now see even greater opportunities to derive value from the cloud as we rise from infrastructure as a service (IaaS), to platform as a service (PaaS) and software as a service (SaaS).
The SAP Web Services platform and the SAP ABAP SDK for Azure described in this paper are PaaS services, offering Azure and SAP developers a streamlined means to connect these systems natively and provide an unprecedented level of functionality and insight into our business processes. In particular:

We’re using our SAP ABAP SDK for Azure to:

- Directly integrate SAP business processes with Azure products such as Azure Key Vault, Azure EventHub, and others.
- Send near real-time notifications to Azure EventHub, where interested parties can subscribe to receive the SAP system status updates. This alert mechanism provides more transparency into the business and for all the involved parties within CSEO.
- Utilize the latest security standards to help secure the entire data exchange.

Further, with our SAP Web Services Platform for Azure, we can:

- Ensure a consistent, scalable, highly available, performant Azure web API layer that exposes the SAP business functionality to internal and external business partners.
- Respond to system and service issues much more quickly with built-in alerts, and then drill into them to gain insight that promotes quicker issue resolution.
- Become more proactive by leveraging the various system health indicators available in Azure tools to head off potential problems before they impact operations.

Our next steps

We’re continuing to enhance the capabilities of our new interfaces. Some aspects we’re exploring include:

- **For the SAP ABAP SDK for Azure:**
  - **Wider suite of supported Azure services.** We plan to extend the Azure SDK for ABAP libraries to include more Azure cloud services.
  - **Functional enhancements.** We’re enhancing the existing libraries with additional functionalities that were also part of other Azure SDKs.
  - **Exploring other opportunities to leverage the SDK.** We’re looking into making the SDK available for other SAP products, including SAP HANA, SAP S/4HANA, and SAP Cloud Platform.

- **For the SAP Web Services Platform for Azure:**
  - **Dashboards.** We’re building automated reports that can be accessed through intuitive dashboards that anyone can use to gain insight into overall system status, health, and performance—a single click will enable users to drill into the details.
  - **Applying Azure ML on telemetry data.** By incorporating Azure Machine Learning (Azure ML), we’ll apply AI to reduce bias from data results and improve the rate at which we receive feedback for product improvement.
  - **Exploring other opportunities to leverage the platform.** We’re looking into extending our platform for use in other SAP products, including SAP HANA, SAP S/4HANA, and SAP Cloud Platform.

Your next steps

If you haven’t already done so, start your SAP journey to the cloud by [creating your free Azure account today](https://microsoft.com/itshowcase). Alternatively, if your organization is on a digital transformation journey similar to the one Microsoft is taking and you are looking for ways to maximize the value of your SAP and Azure investments, we encourage your SAP and Azure development teams to take a closer look at our new SDK and Web Services Platform. With these interfaces, SAP ABAP developers can immediately begin coding with the SDK to reach into Azure; and Azure developers can leverage our Web Services Platform to quickly build an API that works natively with your environment.
For more information

If your organization has an Azure subscription and uses SAP systems, why not extract more value from them both by integrating them with the SAP ABAP SDK for Azure, or by developing your own SAP Web Services Platform for Azure? The following resources provide information about these technologies.

Microsoft IT Showcase

microsoft.com/itshowcase

SAP ABAP SDK for Azure

Redesigning network topology for modern app architecture (This article describes the enterprise hybrid connectivity service that CSEO used to build the Web Services Platform.)

Redesign for the cloud: Building a cloud-first network at Microsoft

Hello Azure: Unpacking how Microsoft moved its SAP workload to the cloud

Strategies for migrating SAP systems to Microsoft Azure

Optimizing SAP for Azure

Building an agile and trusted SAP environment on Microsoft Azure

Create your free Azure account today

© 2018 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.