

## Reading Sample

In this chapter, you'll explore different ways to identify business use cases for SAP BTP. You'll learn about the types of common innovations, see how to identify gaps in your business processes using SAP Activate, and consider your extensibility options.



"Identifying Use Cases"



Contents



Index



The Authors

Smitha Banda, Shibaji Chandra, Chun Aun Gooi

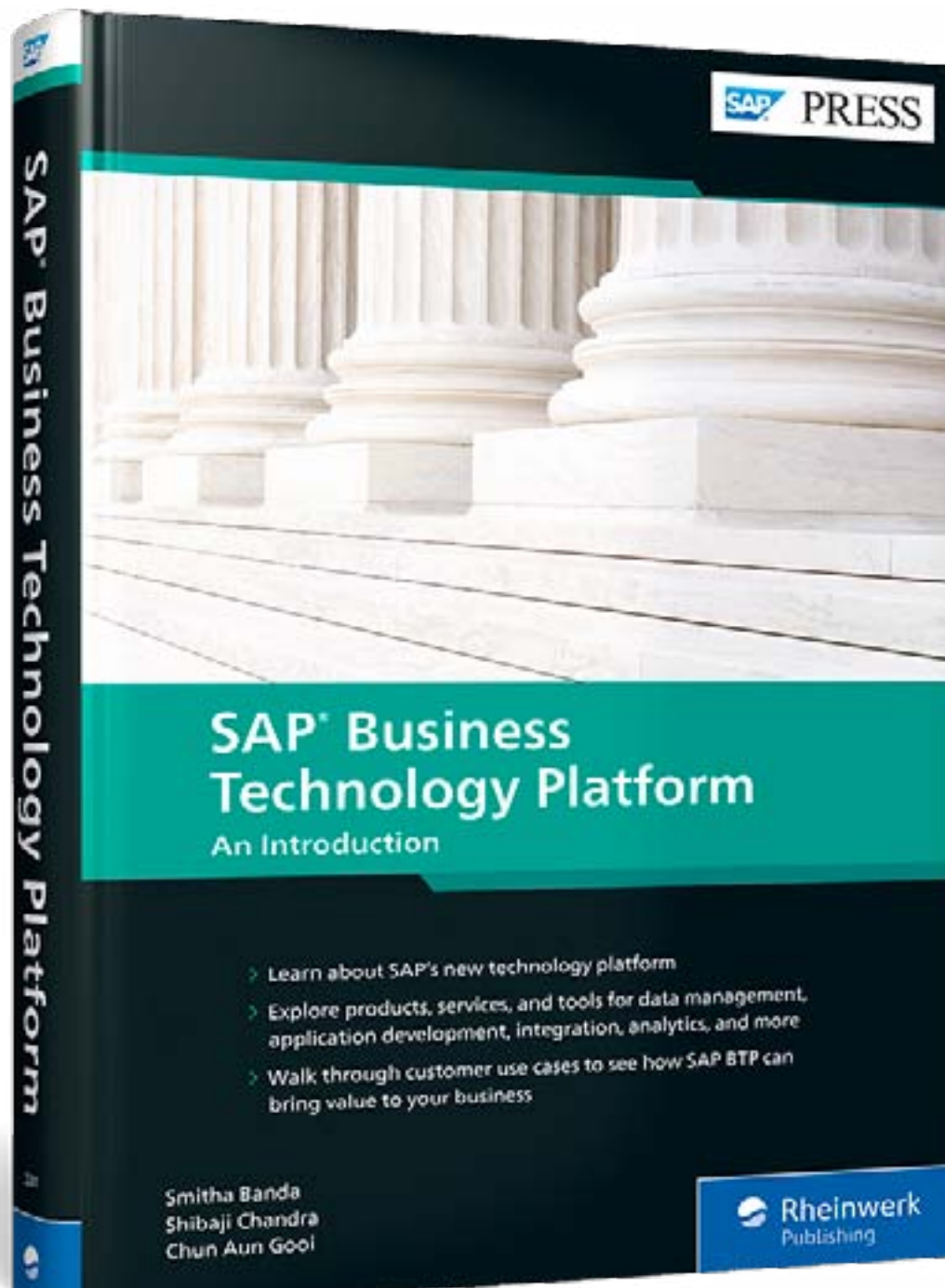
### SAP Business Technology Platform: An Introduction

570 Pages, 2022, \$79.95

ISBN 978-1-4932-2201-8



[www.sap-press.com/5440](http://www.sap-press.com/5440)



## Chapter 11

# Identifying Use Cases

*SAP Business Technology Platform (SAP BTP) offers a range of different capabilities that we've discussed in previous chapters. We also discussed the architectural principles and patterns that are common across SAP BTP applications. Now it's time to go deeper into how SAP BTP is leveraged. In this chapter, we'll look at some use case patterns that will help you identify use cases in your landscape.*

The \$900 sophisticated coffee maker is hissing fumes. The southwest corner of this 10,000 square foot facility is slowly but strongly getting engrossed with the aroma of the finest Brazilian coffee beans. It's a quarter past eight in the morning. Allan, the IT director of Rocking Gas Corporation, is restlessly waiting for his espresso. He has a meeting with Brian, the CTO, in 15 minutes. The email last night had already set the tone. Brian summoned the core steering committee for this urgent meeting to gauge the progress of the digital transformation projects that are led by Allan. One of the key points that surely will be strongly pitched is regarding the value propositions of all the software licenses that were procured in the past few months. Brian is keenly observant of return on investment and value outcomes of his IT investments. Among the different licenses procured, Rocking Gas has also procured SAP BTP in the preceding quarter. Allan personally pursued the SAP BTP adoption as he is well aligned with SAP's overall cloud strategy. Allan knows very well the capabilities of SAP BTP and would like to introduce this new platform within the company's IT landscape more frequently. However, in past months, he hasn't been able to convince line of business (LoB) owners to explore the possibilities. Most of the team members were trying to play it safe. Now, Allan needs to give an explanation of why SAP BTP was underused in its landscape and what can be done to unleash its power to get the best out of the product.

In this chapter, we'll discover the typical use case patterns that customers like Rocking Gas or others can leverage with SAP BTP. We'll then discuss typical business gaps with the SAP Activate methodology and your extensibility options.

### 11.1 Types of Innovation

In the following sections, we'll discuss different areas where business problems can be solved by SAP BTP. SAP BTP provides an open innovation platform that is virtually only limited by your imagination and how you leverage the platform to bring value to your

use cases. To ignite your imagination and for the ease of understanding, we'll describe the following use case patterns:

- Extending functionality
- Extending reach
- Process innovation
- Process automation
- Cross-product workflow
- Cross-functional business processes
- Machine learning
- Internet of Things (IoT)

#### Note

We're in no way stating that these types of innovation use cases are the only possibilities in SAP BTP. Instead, we're focusing on the most popular and trending use cases.

### 11.1.1 Extending Functionality

For many SAP customers like Rocking Gas, SAP ERP or SAP S/4HANA is at the core of their IT landscape managing the enterprise. SAP with its traditional on-premise products gave full flexibility to its customers. Once licenses were procured, customers were virtually free to modify any line of code in the software codebase. SAP has given this immense flexibility for decades. Customers enjoyed this freedom and took the opportunity to shape their business processes as they wished. This high degree of flexibility of course came at a cost. Even though there are certain guidelines and best practices in place, many customers sporadically built their own custom logic to fulfill their business needs. Often these custom implementations aren't well thought out and are too deeply integrated to the core business functionalities. We've seen many customers who didn't explore all the power of standard functionalities offered by SAP in its regular codebase and instead embarked in developing a custom program using the traditional ABAP language. As a result, there has been free-range use of Z programs, custom tables, and user exits. While this gives end customers a lot of flexibility and ease in modifying the core, upgrading SAP to a newer version has become a monumental task. Often, we've seen customers need to take additional attention and budget for an upgrade project. These upgrade projects sometimes go on for months, adding to the total cost of ownership (TCO).

Over the past decade, the software industry has seen a dramatic revolution in innovation. While more and more software companies are coming up with cutting-edge innovations in their products, missing vital upgrades or delaying upgrades is often causing insurmountable costs to customers. As a result, competitors might have gained a vital

edge in this super-competitive market. Cloud-based software as a service (SaaS) products are probably the ultimate solution for this issue. In a pure SaaS world, customers are going to use SAP's standard codebase. In this ideal world, upgrades won't be handled by the customers, but instead managed by SAP, reducing the customer's upgrade cost drastically. This improves speed and agility and enables customers to adapt to market trends. However, with this option, there is limited flexibility to enhance the standard functionality.

SAP has catered to the enterprise needs of more than 400,000 customers worldwide for the past half century. Many have industry-specific solutions. Though SAP has offered a wide range of products in the enterprise core and via industry solutions based on extensive research and continuous innovation, it's often very common that each customer has differentiating processes. While one of the primary benefits of adapting cloud software is to adapt to the standard best practices, you ought to have unique processes that make your business special. These are the invaluable functional enhancements that customers won't let go. While on the one hand, for developers, SAP is putting stricter governance on customization and making options to modify the core fairly limited, on the other hand, for businesses, these differentiating processes are the ones that separate them from their competition. So, we need to find a way to fit these differentiating processes in after the fit-to-standard exercise.

Figure 11.1 shows the concept of extending functionality. This can be achieved by in-app, side-by-side, and hybrid methods, which we'll discuss further in Section 11.3.

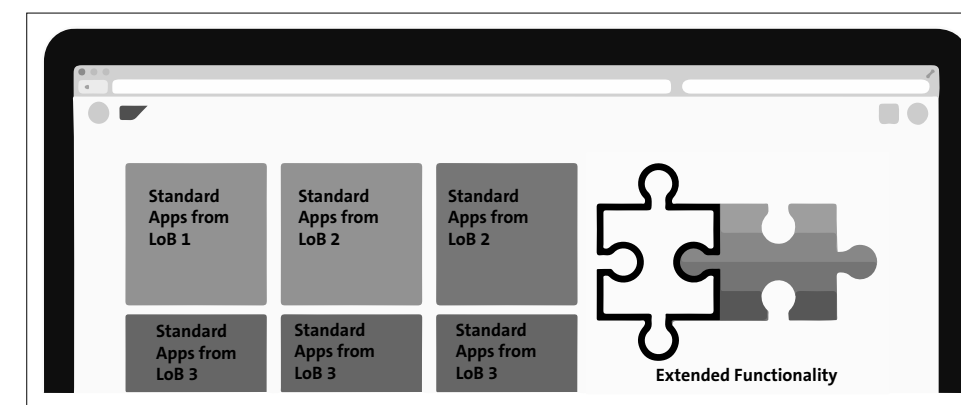


Figure 11.1 Extensibility Concept

During the fit-to-standard analysis, customers might identify add-on functionalities that are required for a business process and aren't exactly covered in the standard software. In such cases, you need to extend the functionality of an application or process. Typically, you start by identifying the functional scope needed here. This might require a user interface (UI) extension, a process extension, or integrating other LoB solutions. Depending on the use case, you may need to adapt an in-app, a side-by-side, or a hybrid

approach. Because the in-app scope is fairly limited, more often we see scenarios requiring a side-by-side extension to be built. A detailed decision matrix will be discussed on whether to go for in-app versus side-by-side extensions in later sections. However, here we'll highlight some of the prominent use case examples of extending functionality using SAP BTP:

#### ■ Industry-specific need

There are standard apps that provide a generic industry-standard solution. However, for a specific industry, you need to add certain fields or logic. This may not be very simple, and the change can't be managed by key user extensibility (which we'll cover in Section 11.3.1). In such cases, there might be a need to modify the existing standard app in combination with an add-on process that is implemented in SAP BTP. It's even possible to create a whole new app using SAP BTP, which has the industry-specific add-on functionality without changing the standard functionality.

#### ■ Customer benefit

Many times, organizations try to use add-on functionalities to make business dealings easier with a specific customer segment. The vital information required to provide the functionality may not be present in the core data model. SAP BTP is the place where you should build the add-on data model, model the add-on process, and deliver the add-on functionality. For example, let's say a company wants to capture geographic location and the corresponding COVID-19 risk data, along with each door-to-door delivery for their products. The traditional enterprise core may not have the necessary data model to fit in COVID-19 risk and geographic information for each customer. Here, you can extend the data model for customers to capture this peripheral data and keep that data in SAP BTP.

#### ■ Government regulations

Many times, country-specific regulations and rules influence the business process. In such cases, if the organization is using a global template for the whole company and wants to make a specific solution for a country-specific app, this can be done via SAP BTP. Browser settings or call routing might be intuitively tracked to redirect to the country-specific app where it's required to have country-specific regulatory steps. Suppose a company is operating in Latin America. There are certain regulatory norms on financial reporting for Latin American countries. SAP BTP provides standard content for integration with such government authorities so that you need not worry about these additional regulatory requirements.

#### ■ Compliance and standards

Though SAP has a wide coverage of various compliance and standards across the world, there might be specific compliance or standards that the customer organization needs to adhere to and that the standard applications aren't fully complying with. In such cases, you can extend the functionality in SAP BTP to comply with these standards. For example, any company operating in the European Union has to comply with the General Data Protection Regulation (GDPR). SAP BTP is compliant with GDPR, so you need not worry about putting out any additional effort to achieve compliance.

### 11.1.2 Extending Reach

The modern world is complex. With wide internet coverage across the world, mobile network connectivity, social media, and IoT, we have unprecedented network growth. We've spread a multiplanetary network across our solar system that can send communications in matter of a blip. More than 7 billion people on this planet are connected for their day-to-day work. Through smartphones and 5G connectivity, we've seen exponential growth of connectivity between humans and machine users. Fifty years ago, when SAP started its journey, there was barely any email communication across the world. Today, more than 3 billion people are connected over social media and sharing the moments of their lives every second, not only via text messages but also with a relentless infinite stream of media information flow.

Accordingly, the extent of business has dramatically changed over the past decade. Companies are no longer relying on conventional intercompany message exchanges such as Electronic Data Interchange (EDI), supply chains are managed via sophisticated business networks such as SAP Business Network, and business partners are collaborating and negotiating contracts over more interactive social media style platforms. Over the past decade with the growth of the internet and the progress of cloud technologies, we've seen newer offerings from SAP and dramatic improvements over SAP's architecture. From the monolithic architecture-based SAP R/2 to the three-tier SAP R/3 to the highly scalable, agile, microservice-based, cloud-native modern software service offerings, SAP has evolved a lot.

Figure 11.2 shows the extending reach concept. As you can see, the core functionality is extended to different channels, such as the internet and mobile devices.

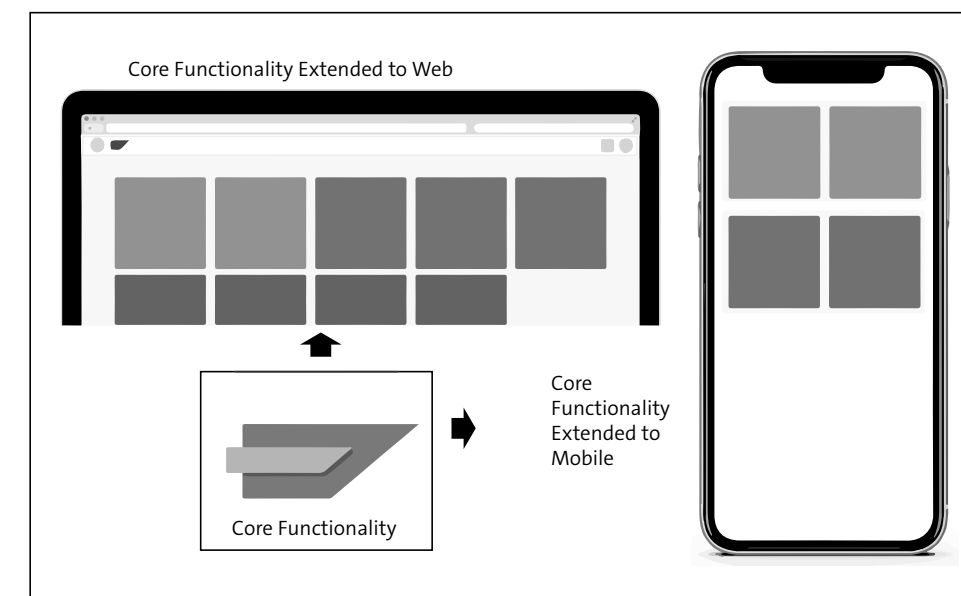


Figure 11.2 Extending Reach



Because SAP BTP is offered on the cloud, its reach is wide and far extending. First, the platform is on the open internet. So, connecting to the world, be it business-to-business (B2B), business-to-consumer (B2C), or business-to-government (B2G), can be executed with extreme ease. No more difficult network system management, network zone management or complex firewall rules are necessary. You've got an open, secured platform on the public cloud to interact B2B, B2C, and B2G. Let's explore the typical use case patterns when a business wants to extend its reach:

#### ■ Adding channels

Many times, businesses want to simply add different channels to increase reach to their business. For this case, it's not necessary to enhance the core functionality. For example, an organization uses a time sheet application that lets its employees fill in times every week. Traditionally, the SAP time sheet would be used via Transaction CATSXT. This was good for in-house employees who are working well within the office network, but there could be other employees who aren't working within the corporate network, so a web interface would be more useful. SAP Fiori apps can be used via SAP BTP central launchpad in such cases. Again, this can further be extended to mobile or tablet devices for employees who are working in the field.

#### ■ Providing collaborative platform

Collaborative platforms such as SAP Jam or Microsoft Teams are gaining popularity in business every day. There are growing trends to provide platforms to your business partners to exchange ideas, negotiate contracts, and engage in real-time collaboration to grow business. SAP Work Zone on SAP BTP is a perfect solution that provides such services. You can use SAP Work Zone to bring in a nice intuitive digital experience that provides secured role-based access to various business partners. This can be a platform for exchanging notes and documents, doing transactions, viewing critical analytics, and managing cross-business workflow.

#### ■ Internal versus external user base

We've seen many use cases where a business wants to expose part of its core business functionality to some wider external users. These users may not be part of this core organization, such as contractors, suppliers, customers, or any third party. SAP BTP provides a nice segregation between your core and the external-facing layer. This is where the two-layered architecture comes into play. External users can use vital information and do critical business transactions using various channels such as the SAP Launchpad service, SAP Work Zone, or mobile devices using SAP Mobile Services or even interact with artificial intelligence (AI) conversational bots built on SAP Conversational AI.

#### ■ Extending to government service

Government regulations are part of business, and businesses need to comply with such requirements all the time. In the past, we've seen businesses develop custom solutions to comply and integrate with such government data interchange. With SAP BTP business services, you can leverage many standards that help organizations with their B2G interactions.

#### ■ Online and offline scenarios

There are certain critical situations where certain users may not be able to get online access to the system. A typical example is a plant shop floor with hazardous materials where there are restrictions on connectivity and networks. In the past, either a manually intensive process needed to take place to record business data in such situations, or, in some cases, costly compliant devices needed to be installed. With SAP BTP mobile development kits (MDKs), you can now build business apps with offline support in a matter of a few days.

#### 11.1.3 Process Innovation

Consider a brick-and-mortar grocery shop that used to have two employees. It was the norm for all customers to stop by the shop for retail groceries. The two employees who supported the business were receiving a fixed salary. The shop was open Tuesday to Sunday, 10:00 am to 8:00 pm. Usually, local customers couldn't get any service outside this stipulated time. However, during the COVID-19 pandemic and lockdown, customers couldn't reach such shops, resulting in a sharp decline in their business. Within a month, this small vendor opened a door-to-door service serving the neighborhood for an extended range of time. On demand, the business process was remodeled almost overnight. There are additional employees now who are working on a commission basis for a wider time range, and they have employed one employee to handle phone call requests in the store and one employee to manage inventory.

This example emphasizes the need for flexibility and agility to do business in the fluid world of today. This situation is no different for Fortune 500 companies.

SAP BTP really comes in handy in such situations where managing complex business processes have never been easier. The following lists some of the typical use case scenario patterns:

#### ■ Defining a process model

Customers can use business process intelligence tools such as SAP Signavio to model business processes. SAP Signavio is a powerful tool with which you can define the end-to-end business processes by using an all-in-one business process management journey. These process models are purely driven by business requirements. At this stage, the customer may not identify which part of the process is covered by standard development and which part requires custom development.

#### ■ Using process templates

SAP BTP's process flexibility capability is part of SAP Intelligent Business Process Management (SAP Intelligent BPM) suite. This service offers standard business processes out of the box that can easily be adopted and modified according to business needs. Thus, it's not always required to build processes from scratch; rather, you can quickly modify and adopt a standard process.

### ■ New process outside core

Sometimes you may have an innovative process that isn't related to your core business process. You may need new edge solutions such as machine learning or augmented reality for such processes. SAP BTP can be your innovation platform in this case. You can leverage machine learning foundation from an SAP BTP AI service or even leverage a host of AI-driven business services such as document classification, document extraction, and so on to define and manage your process.

### ■ Process integration

Often, you'll find the need to integrate your innovative process with some of the established processes in the core. Integration to this core process is the key here to succeed to bring in the overall value to business. In certain cases, SAP BTP offers SAP Integration Suite. At the heart of SAP Integration Suite, you'll find the Cloud Integration capability that provides a means to connect, map, and orchestrate message-based process flows. You can use a host of standard adapters to connect a wide range of solutions. If needed, you can develop your own adapter as well. For all practical purposes, however, Cloud Integration does provide a robust framework and wide range of adapters that are adequately sufficient for most scenarios.

#### 11.1.4 Process Automation

Let's recall a famous scene from *Modern Times*, a silent film from the 1930s by Charlie Chaplin. The worker in the assembly line was twisting the bolts, perhaps a million times a day. Perhaps those scenes were a reality during the early 20th century. However, during the fourth industrial revolution, we've seen robots replacing human labors for repetitive tasks. Now if you go to any modern-day shop floor, you'll hardly find people twisting bolts in an assembly line. Major car manufacturers and heavy industries have put sophisticated robots to do such hazardous and repetitive tasks. Human effort, however, is leveraged in what they do best—design and innovation.

Robotic process automation (RPA) is providing the same value proposition today as the industrial robots provided to the production assembly line. Imagine in your day-to-day office work how many repetitive tasks you must do. Perhaps you're a procurement manager, and you need to find the best price of various commodity goods required in your organization. You need to go to various websites and collect price listings, carefully examine various criteria, and find the best match. Or suppose you're a car salesman. Every day, the dealership provides the list of cars arriving in the lot, and you need to put the car specs, price, offers, and margins in the system and provide a detailed daily report on sales and inventory balances to your sales manager. This is just a glimpse. In the real world, there are many repetitive tasks we have to do in our daily life, which perhaps don't make the best productive use of our potential. You can argue that we have APIs and other sophisticated means to integrate applications, and targeted business applications help business users ease laborious processes to execute their

tasks. However, in reality, we still need to click, copy, and paste in a repetitive fashion too many times every day.

Figure 11.3 shows the value proposition of intelligent processes that assist business users, help adapt processes, and automate processes.

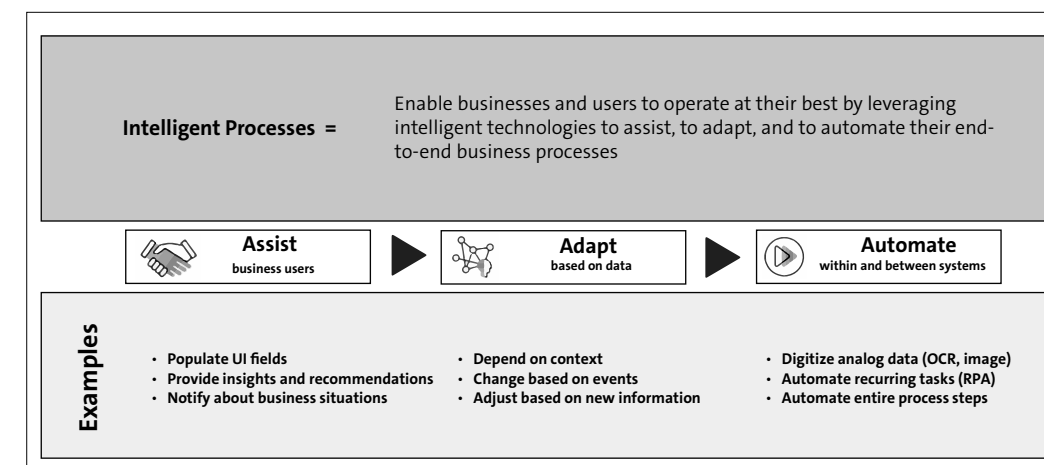


Figure 11.3 Process Intelligence

SAP Intelligent Robotic Process Automation (SAP Intelligent RPA) is aligned with its intelligent process strategy. Broadly, SAP Intelligent RPA accelerates the digital transformation of business processes by automatically replicating tedious actions that have no added value. It offers the following modes:

### ■ Attended mode

RPA can be semiautomated where the robot interacts with the user to execute the tasks. In such scenarios, the bot executes certain tasks but may need human assistance or guidance. This scenario is also referred to as robotic desktop automation. Typically, the scenarios covered in this type of RPA include screen scraping, website or GUI-based applications, and so on in which bots co-work with human users.

### ■ Unattended mode

In this RPA mode, the bot is completely doing the task on its own, maybe in background in such cases. These processes are often scheduled, highly scalable, and use APIs and script tasks.

Figure 11.4 shows the different components of SAP Intelligent RPA. SAP Intelligent RPA has an agent typically installed in a local computer. You need to design the process flow in the designer that is hosted on the cloud. The automated process can be orchestrated based on your process design and can use the agent for local processing.

SAP BTP provides a full-scale, full-range service with SAP Intelligent RPA. Through this service, you can train bots to do the repetitive tasks, manage environments, and even

integrate the automated bot processes to other SAP BTP services. For example, you can initiate an automated bot-based process from within a chat interaction via SAP Conversational AI. Again, it's very possible to integrate other SAP BTP services in a bot-based process flow. For example, an unattended bot may be checking incoming email requests and then sending the email body to the SAP BTP machine learning service to do text extraction. If the bot finds a request to quote, then it calls an API via SAP BTP API management to get quote information from SAP Customer Experience, saves the record on SAP BTP's SAP HANA Cloud service, and sends an email notification to the requester.

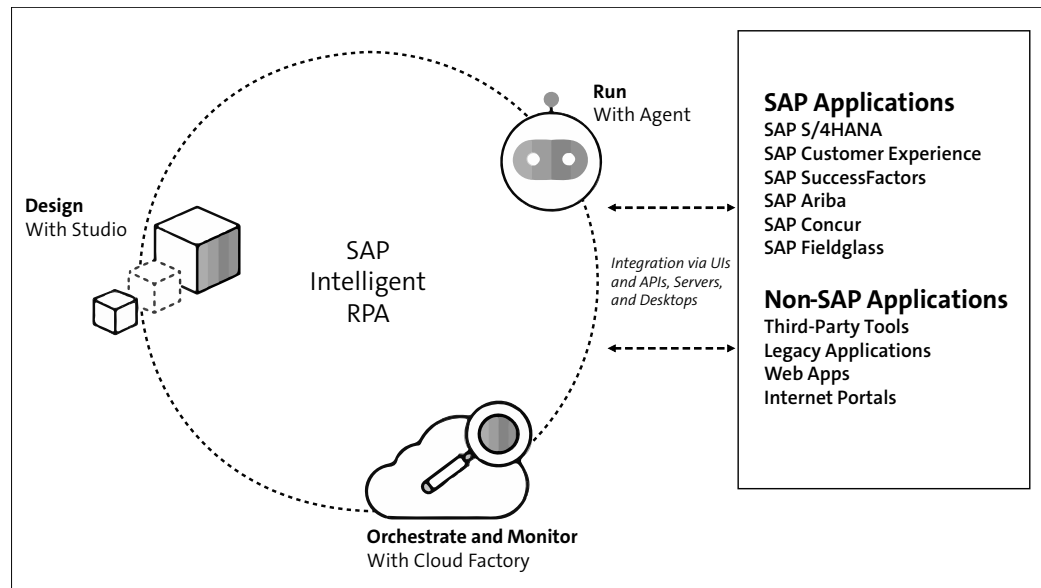


Figure 11.4 Components of SAP Intelligent RPA

### 11.1.5 Cross-Product Workflow

Over the past several years, enterprise applications have become more complex. The extent and functionality coverage has increased. As a result, enterprises are relying on specialized LoB solutions more and more, which is resulting in a distributed architecture. For example, when your organization was small, you could have managed all your core business operations, such as customer experience, sales and distribution, procurement, production, human resource management, and finance, all in a single core system (e.g., SAP ERP). Now, as your organization is growing and extending wide in functionality and reach, you may want to distribute certain specialized operations to specialized LoB solutions. For example, you're now managing HR operations in SAP SuccessFactors, customer engagement is via SAP Customer Experience, supply chain via SAP Ariba, travel operations by SAP Concur, and the list goes on. However, at the heart of the entire landscape, you still need some core solution such as SAP ERP or SAP

S/4HANA to manage your core business operations and processes. In a modern-day enterprise, this kind of distributed landscape is more common. Often, we see processes that touch multiple LoB solutions. This is where a cross-application/cross-product workflow comes into the picture.

Figure 11.5 depicts the concept of a cross-product workflow. One workflow may touch different business objects from different LoB solutions.

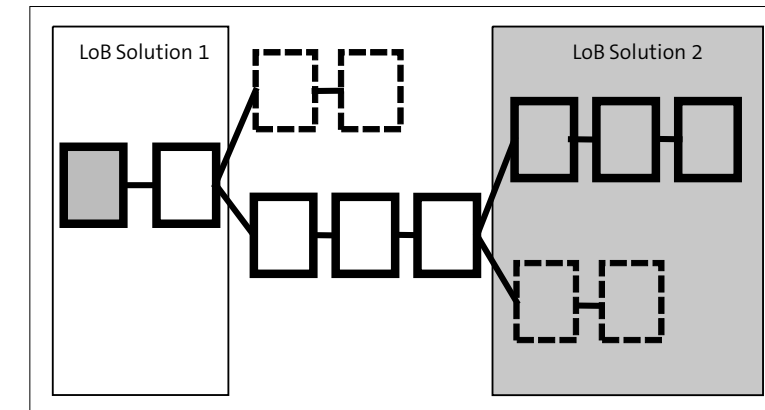


Figure 11.5 Cross-Product Workflow

In many SAP implementations, different LoB-centric solution teams work in silos. They are more focused on their respective areas and try to provide solutions to the enterprise in their own way. This is where enterprise architects and project management should step in to see the bigger picture. It's now more necessary to bring in the integration questions earlier in the implementation cycle. This is where you need to bring in a wider view to see the overall picture rather than looking deep into one LoB-centric solution. For example, we've seen enterprises implementing SAP SuccessFactors, SAP S/4HANA, and SAP Concur in three different projects with three different teams. Often, there is no or little harmonization between these project teams. This results in some blind spots for your enterprise. We've seen enterprises investing in these projects in silos for too long and going into narrower objectives. Because you're not viewing the solution from the perspective of the entire organization, it's very common to miss out on opportunities.

Another pitfall here is the tendency to focus on existing workflows. In a centralized system, you may have some existing workflow cutting across different business objects or audiences. In a to-be world, you need to see the overall scenario in the enterprise before you decide on how to go with such a workflow. For example, you may have a travel approval workflow built on SAP Business Workflow in SAP ERP. Your organization is embarking on an SAP S/4HANA transformation journey, and you want to reimplement the workflow now in SAP S/4HANA. However, before you jump onto the technical nitty-gritty of how to reimplement this workflow within SAP S/4HANA, check out

whether the travel operations are still going to be done entirely within SAP S/4HANA or whether SAP Concur is being implemented by another project team that may need to be integrated in the near future. If the entire solution is now distributed over multiple products (here, SAP S/4HANA and SAP Concur), it would probably make more sense to use a cross-product workflow.

To help with this process, SAP BTP comes with scalable, flexible, full-scale workflow management that is based on open-source technologies, as shown in Figure 11.6. SAP Workflow Management provides not only a flexible suite to manage your workflow but also a solution to manage business rules, which are vital inputs on how the workflow will behave contextually. In addition, it comes with a process visibility dashboard that shows real-time information on how the workflow instances are behaving. You can show intuitive information based on key performance indicators (KPIs) to plug in to vital business statistics that will drive your actions through the workflow. In addition, an integrated inbox called the My Inbox app is part of the suite that manages all your workflow tasks in a more modern and innovative UX. The tile-based UX for process visibility shows your KPIs, provides analytics, and helps you manage the overall process flow across your enterprise.

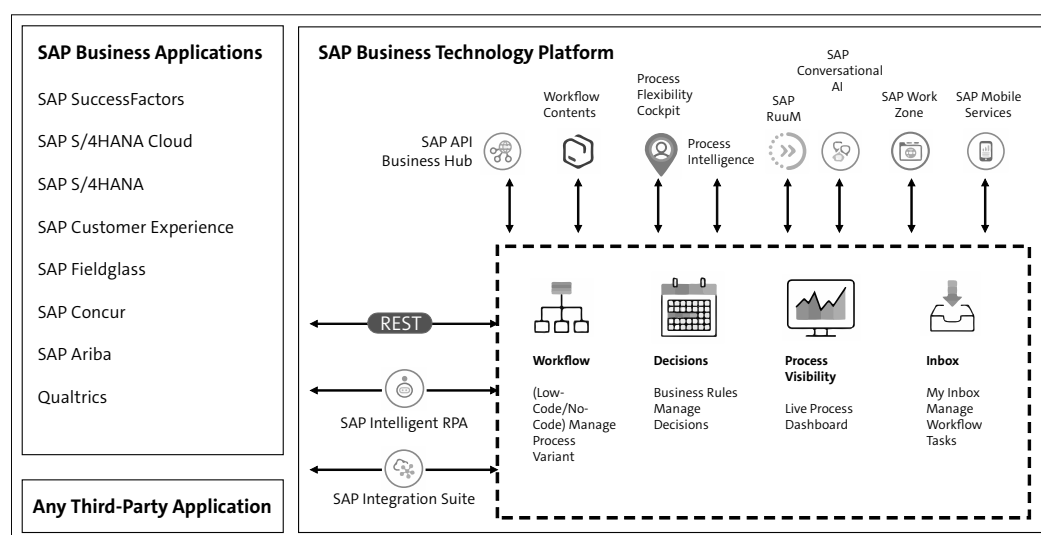


Figure 11.6 SAP Workflow Management

Certainly, you'll discover many possibilities in SAP Workflow Management that were far into your imagination in a traditional workflow world. As Figure 11.6 shows, SAP Workflow Management offers different capabilities such as workflow, business rules, process visibility, and the My Inbox app that can be integrated into your SAP BTP launchpad to centrally manage all tasks and KPIs. This workflow sits in the middle of your landscape and connects processes across different LoB solutions.

### 11.1.6 Cross-Functional Business Processes

When the horizon of your enterprise expands, the functionalities and end-to-end processes can barely be catered to by one product or solution. Often, we see complex process flows such as procure to pay or quote to invoice spanning multiple systems—simply because in a modern distributed landscape, you have specialized LoB solutions spanning landscapes that cut across various parts of the processes.

Let's consider, for example, an organization managing the procure-to-pay process. In the past, if you had a simple core system, you would probably be content with SAP's procurement and finance modules. You would manage your vendors, prepare contracts, place a purchase order, manage inbound delivery, and eventually do the due diligence on financials by managing accounts payables. However, more possibilities are now on the table. Your competitors are negotiating with vendors to get lower prices and better quality every day, eventually increasing profit margin and outperforming your organization. So, the extent of the procure-to-pay process doesn't start from vendor onboarding anymore; you probably need a better platform to negotiate contracts with vendors, provide ease in collaboration, and offer a guided buying procedure. For all these cutting-edge business processes, SAP offers SAP Ariba functionalities. However, now even if you move your procurement process to SAP Ariba, you still need financials reporting to the core. As a result, the process is now extended across your ERP core (SAP ERP or SAP S/4HANA) and SAP Ariba.

As we've seen, end-to-end business processes may span different LoB solutions, which is known as cross-functional business processes, as shown in Figure 11.7.

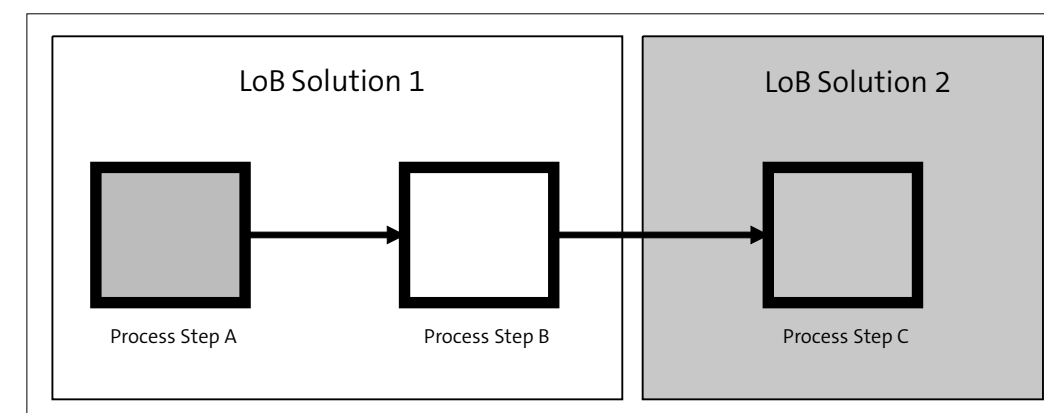


Figure 11.7 Cross-Functional Business Process

Cross-functional business processes have the following requirements:

- **Domain model**

Cross-functional business processes must be integrated across the products. SAP BTP provides a one domain model for the standard business objects for such end-to-end



processes. In addition, it provides SAP Graph to explore the details of the domain model that can be leveraged to integrate these systems seamlessly.

#### ■ Integration need

Though SAP provides one domain model for most critical business objects and entities, and is expanding fast in this area, you may encounter various scenarios where you need specialized integration, custom mapping, and message orchestration. In these cases, the Cloud Integration capability comes in handy. The Cloud Integration capability provides full swing capabilities to integrate different systems on an application-to-application (A2A) or B2B integration.

#### ■ Cross-LoB functionality

There might be scenarios where a business is innovating some process that requires integration between two processes in two different LoBs. For example, suppose you're trying to make an agile approval process for your purchase requisitions. Apparently, this process is tied up to procure to pay, and it may seem that all the necessary business entities are present in SAP S/4HANA itself. But in a distributed landscape, you may have the vendor information from SAP S/4HANA, product catalog coming from SAP Ariba, and organizational hierarchy from SAP SuccessFactors. In such cases, you typically need a common platform where you'll build your business logic and give users a place to operate. SAP BTP is the right place as it provides a cross-product workflow and SAP BTP launchpad that integrates the inbox, analytics, and vital KPIs in one place. You can manage all business users via the Identity Authentication service and give them role-based access to perform their business operations.

### 11.1.7 Machine Learning

AI is now part and parcel of our daily lives. When we see movie recommendations from Netflix based on our likes or dislikes, auto complete in our email applications, and driverless autopilot in action on the streets, we're getting service from AI. Every like or reaction to your Facebook post is fed back to contribute to shaping your personalized content.

Machine learning is the backbone of AI. Instead of the traditional approach where we used to tell computers the algorithm to compute data, it's the reverse here. We feed in real data to come up with mathematical models that are best fit to see patterns or trends or to solve a particular problem. So, we start with data and arrive at an algorithm that best addresses the pattern of the problem. This is immensely useful for business use cases. We're now seeing scenarios where, based on sales history data, we can predict the sales forecast more accurately, use regression algorithms to find patterns on commodity price trends, and apply clustering algorithms to better group customer segmentations. These are just the tip of the iceberg. Implications of machine learning algorithms in business processes are deep and far reaching.

Figure 11.8 shows the type of machine learning and typical example use cases. It shows the typical classification of different machine learning algorithms. Broadly, we can classify machine learning algorithms into four categories:

#### ■ Supervised learning

In this type of machine learning, the machine learning algorithm maps an input to an output based on some example input-output (I/O) pairs. In other words, it has labels. Here are a few examples:

- Regression algorithms can be used for prediction or trend analysis, for example, to determine future sales volume or trends of a product popularity.
- Classification algorithms can be used to classify cases, for example, to determine whether a purchasing transaction is a fraudulent one based on time, amount, location, and so on.

#### ■ Unsupervised learning

For unsupervised learning, no labels are provided as input. Various popular clustering algorithms are used such as Random Forest, K Nearest Means, and so on. These types of algorithms are helpful to customer segmentation, for example.

#### ■ Semisupervised learning

In this type of machine learning algorithms, some sets of data are provided with label inputs and other sets of data can use that input to cluster the other broader examples.

#### ■ Reinforced learning

This type of machine learning algorithm is more popular for self-learning autopilot and game programming where the model learns and improves itself based on the reward.

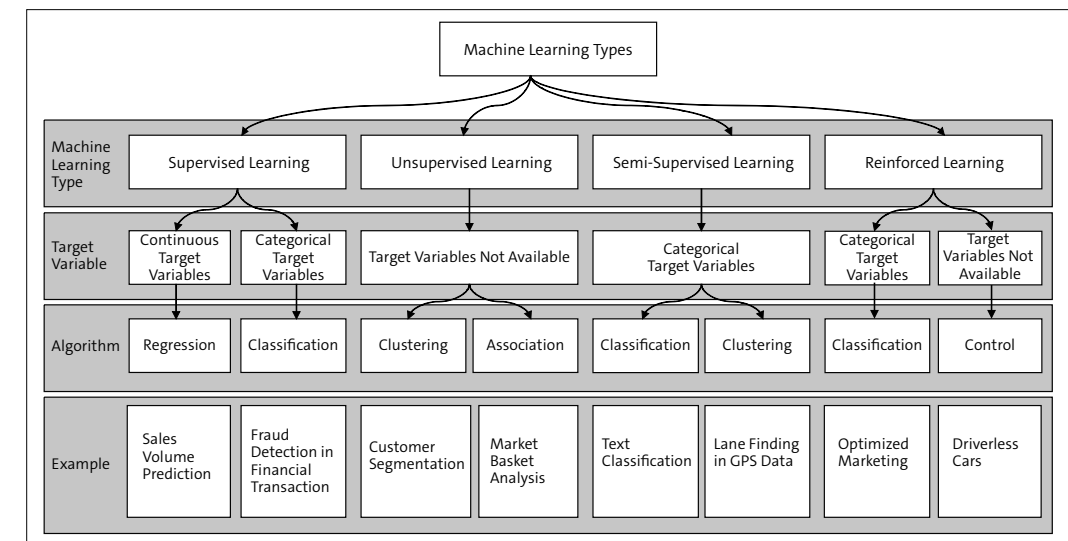


Figure 11.8 Machine Learning Types and Examples

SAP has already invested heavily in research, developing algorithms, and providing fine-tuned business services that leverage full-scale machine learning. There are primarily three ways SAP offers machine learning and AI capabilities:

- **Embedded**

Many machine learning algorithms are now embedded in standard functionalities of SAP products. For example, the automatic account reconciliation functionality in SAP S/4HANA uses machine learning, as does SAP Concur to extract vital information out of scanned receipts and automatically post the information in the expense report. It's beyond the scope of this book to discuss the extent of these embedded intelligent applications, but we want to emphasize that within SAP products, there is already a large machine learning footprint.

- **As a service**

SAP also provides the platform and services to build, modify, and leverage machine learning-based business functionalities in the form of services. This is exactly where SAP BTP comes into the picture. You may have scenarios where you need to classify documents or extract vital information from scanned documents. SAP AI Business Services provides you standard functionality to extract the most commonly used fields from a list of business documents, such as invoices. You might have to spend months to develop such an algorithm that is fine-tuned for your scenario. With SAP AI Business Services, you can do this in just a few hours. It delivers preconfigured machine learning models, allows you to train the models based on your data, and then provides an inference layer API so that you can consume the result based on an open architecture.

- **Developing platform**

Often the primary challenge to develop meaningful and effective machine learning models isn't the skill set or understanding of mathematical models, but instead finding good data. A lot of data is required to develop and train models, and the quality of the data must be maintained. For example, for an image recognition problem, you may need to feed the machine learning algorithm with thousands of sample training images with consistent resolution and size. Often resizing and cleansing the data in the right format is a challenge. Again, it's not always good to have all positive use cases to train a model. You need to have a good mix of various data to avoid overfitting problems. Yet we need a platform to try and experiment with various machine learning algorithms to play around and see which is best suited. The platform needs to provide robust capabilities like large-scale data ingestion, data cleansing to maintain quality, and flexibility to explore new paths and find the best one.

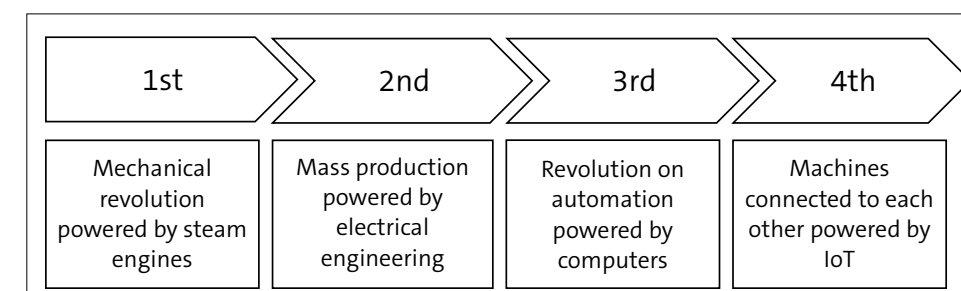
SAP BTP provides all these capabilities with SAP Data Intelligence, where you can do data ingestion from various data sources, merge and wrangle data, cleanse according to need and maintain high quality, and have a flexible framework to use Jupyter

Notebook to experiment with machine learning algorithms such as deep learning, convolutional neural networks, and so on. In addition, SAP Data Intelligence provides the means to publish the inference layer of the developed model via APIs so that you can easily consume the outcomes from your business applications.

### 11.1.8 Internet of Things

They say the first industrial revolution was the use of steam engines in industries, reducing manual mechanical labor. The second revolution was powered by electricity and scientific progress. The third revolution emerged with unprecedented progress in computer science and telecommunications. We're now at a stage to embark into the fourth industrial revolution powered by IoT. The internet, a network of networks, was originally meant to connect people with other people. Of course, these communications were possible by means of machines. There is a host of electronic equipment that made the internet possible. The fundamental shift of thought in IoT is that communication is initiated and received between machines.

Figure 11.9 shows the different stages of industrial revolution: mechanical, electrical, automation, and IoT.



**Figure 11.9** Stages of Industrial Revolution

During the past decade, we've seen many sporadic solutions to connect industrial machineries to computer systems, including efforts to integrate machine statistics to feed into SAP Manufacturing Integration and Intelligence (SAP MII) solutions. However, what IoT brings into the industry is vaster and wider. This isn't only limited to specific mechanical devices or specific industries. IoT is based on certain industry standards. Manufacturers from consumer goods to retail to heavy industries everywhere are trying to set these standards, and machine-to-machine communication protocols are standardized. This results in wider possibilities to provide a platform that enables customers across various industries to bring in digital twins (the digital replica of a physical machine or object that has similar characteristics as the physical object). Unlike previous solutions to address specific problems, IoT demands a wider platform

that enables customers to manage devices, process messages, and handle a wide variety of low-level edge protocols.

SAP BTP provides SAP Internet of Things (SAP IoT), which fulfills all these needs. Broadly, SAP IoT provides the following:

- **Digital twin model**

You can model a digital twin of your device in SAP IoT to help identify performance, emulate features, and model control behaviors.

- **IoT-driven event response**

SAP IoT provides out-of-the-box support for various edge protocols (the low-level protocols for machine-to-machine communication) to support various industry standards. The SAP Edge Services platform supports various protocols such as Constrained Application Protocol (CoAP), File, Modbus, Message Queuing Telemetry Transport (MQTT), Open Platform Communications United Architecture (OPU CA), representational transfer (REST), SigFox, and Simple Network Management Protocol (SNMP).

- **Auto ID–based smart sensing**

Identify things in real time using smart sensing and integrate them into core business processes.

- **IoT data lake**

SAP IoT provides a robust, state-of-the-art, customer-managed data lake solution to manage multitier time series data.

- **IoT-empowered business process**

Reimagine your business process to explore possibilities by incorporating IoT data. For example, large agricultural farms are now using various sensory data such as soil moisture, water level, ambient temperature and so on to control watering plants in the farm fields. Drone images are being used in combination with machine learning to recognize color patterns of crop fields to understand the growth of crops in specific fields.

- **IoT analytics**

You can use the SAP BTP cockpit to get standard views on devices and message behaviors. In addition, these vital statistics again can be conglomerated with other business data to provide more insights using SAP Analytics Cloud.

## 11.2 Identify Business Gaps

As SAP is focusing more offerings on cloud-based software, more and more functionalities are becoming standard. In addition, systems are upgraded to the latest and greatest to match up with all the latest innovations more easily.

At least, that's the intention. However, we know SAP landscapes for customers aren't so simple. There are modifications and key differentiating processes that may not be possible to cover within the standard functionality. Unlike traditional project management methodologies, in SAP S/4HANA and SAP S/4HANA Cloud projects, we rely on the SAP Activate methodology, which emphasizes a *fit-to-standard* approach.

Figure 11.10 shows the significance of the fit-to-standard exercise in a typical SAP S/4HANA project where the SAP Activate methodology is followed. We discover the initial scope in the discover phase, and the business-driven configuration occurs in the prepare phase. In the explore phase, the fit-to-standard workshop tries to map the business processes and fit it to the standard delivered processes; as a result, a configuration backlog is produced. In the realize phase, these backlogs drive the necessary configuration and development needs. One of the primary deliverables for the fit-to-standard workshop is configuration backlogs and integration and extension requirements. This results in a list of identified functional gaps that translates into interface and extension requirements.

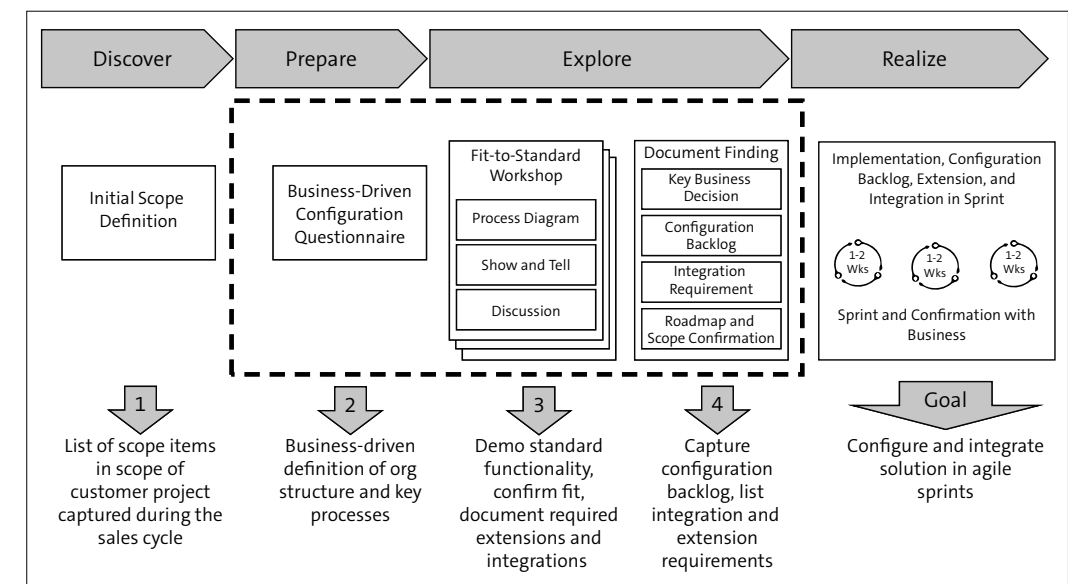


Figure 11.10 SAP Activate Methodology and Fit-to-Standard Exercise

On a broad level, the overall approach to implementation projects has changed. Figure 11.11 shows the difference in development approaches between past SAP ERP and SAP S/4HANA implementations and modern SAP S/4HANA implementations. As you can see, in the past, you have a license for the ERP system, and it's installed on-premise. As a result, customers have full freedom to modify the ERP from core code. Typically, these enhancements are very tightly coupled with the core solution. The downside is

that this tightly coupled enhancement often breaks the upgrade path. Of course, developers enjoyed the homogeneous technical landscape where virtually the entire project was executed on the ABAP platform. For many implementations, this is confined within a few homogeneous solutions such as SAP ERP, SAP Customer Relationship Management (SAP CRM), SAP Human Capital Management (SAP HCM), and so on. However, there was a trade-off with scalability. We needed sophisticated custom solutions and ingenuity to scale a solution. Overall, the TCO for the solution and solution upgrade was much higher.

In modern SAP S/4HANA implementations, the approach is quite different. After the fit-to-standard workshop, we're now able to understand the functional gaps. These functional gaps are often listed in key design documents (KDDs). Then one functional gap may translate to one or multiple collections of interfaces and extensions. There are well-defined methodologies to identify and approach these extensions and classify them to be implemented via an in-app, side-by-side, or hybrid approach (see Section 11.3). The side-by-side extensions are loosely coupled to the core solution based on a stable contract to whitelisted APIs. Side-by-side extensibility gives flexibility in terms of a wide range of technology choice. You can leverage the full potential of open-source technologies to implement such extensions. As SAP BTP-based side-by-side extensions are cloud based, these are inherently scalable. SAP BTP provides first-class tooling such as SAP Web IDE, SAP Business Application Studio, or Eclipse for ABAP Development Tools (ADT) that developers can leverage to quickly build SAP applications. So, for this modern approach, there are shorter sprint cycles to manage projects, and value-based outcomes come in shorter cycles. We've discussed the DevOps practice and value of continuous integration and continuous delivery (CI/CD) in Chapter 10. Efficient DevOps and CI/CD practices help in such implementations and provide quicker time to value.

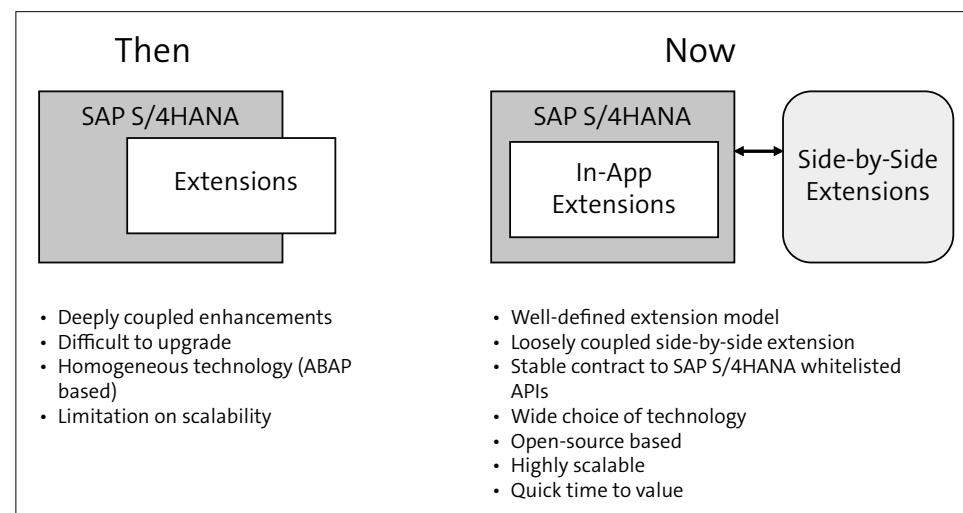


Figure 11.11 Approach to Extensions

## 11.3 Extensibility Concept

Unlike the traditional enhancement options such as user exits, enhancement spots, and so on, SAP S/4HANA suggests doing the functional enhancements by extensions. There are primarily two techniques that we follow to extend a SaaS solution, which we'll discuss and compare in the following sections.

### 11.3.1 In-App Extensions

SaaS solutions are typically delivered in the cloud, so they share a codebase between different tenants. When you get a subscription to a SaaS solution, you don't have full control over the codebase. In addition, the SaaS provider can't provide full customization specific to your needs as one specific customization may contradict others. So, SaaS products provide a controlled mechanism to implement some part of an extension. There are provisions to extend the data models. In addition, there are specific points in the UX where you can plug in extensions. For example, SAP SuccessFactors provides the Metadata Framework (MDF) to enhance the data model. In addition, you can add custom tiles and pages in SAP SuccessFactors apps.

SAP S/4HANA provides certain mechanisms that allow power users to enhance the data model, build custom logic, and build the UI on top of it. This is called *key user extensibility*. Figure 11.12 gives some more insights on different in-app extension mechanisms that SAP S/4HANA provides.

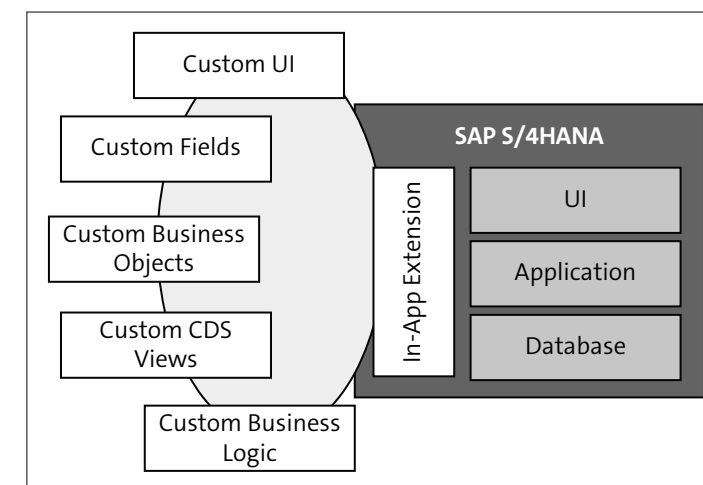


Figure 11.12 SAP S/4HANA In-App Extension Mechanisms

Key user extensibility mechanisms can be categorized under the following, as shown in Figure 11.12:



■ UI adaptation

Figure 11.13 shows UI adaptation possibilities by key users in an SAP S/4HANA system. When you click the **Adapt UI** option, you can do the following:

- Hide fields in a form or table and hide groups and areas.
- Rename labels.
- Add fields to the UI from the field repository.
- Move form fields to UI groups.
- Combine or split fields.
- Define new filters or variants.

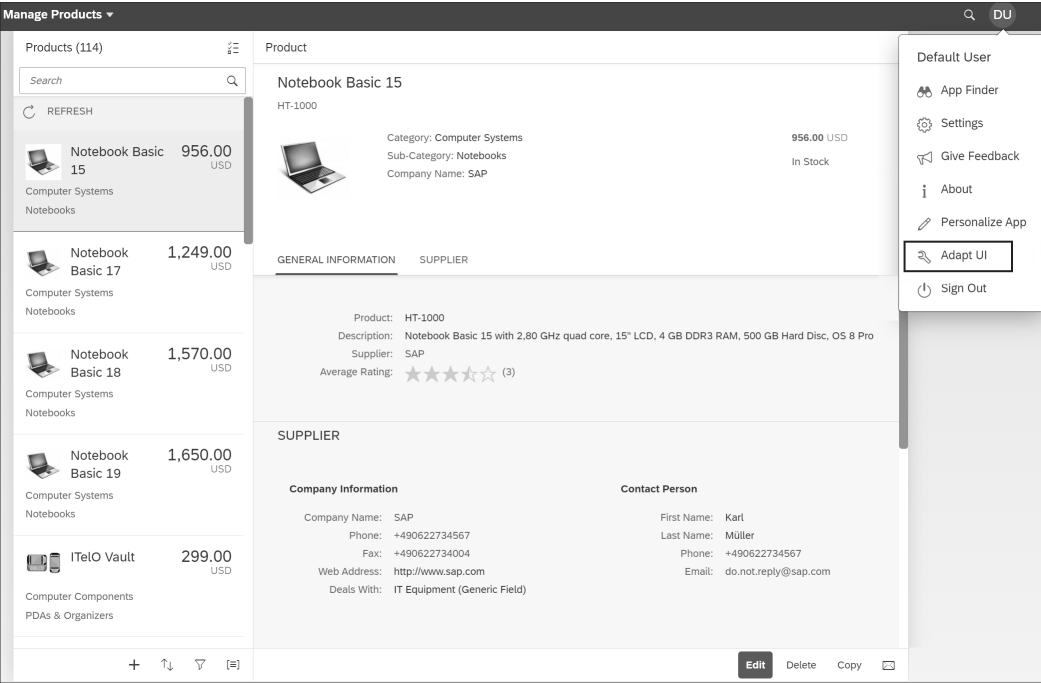


Figure 11.13 UI Adaptation Options by Key Users

■ Custom fields and tables

Figure 11.14 shows the Custom Fields and Logic app, where you can create custom fields in SAP S/4HANA. The following options are available:

- Create and edit custom fields.
- Delete existing custom fields.
- Manage usage of custom fields.
- Define associations.
- Extend CDS views and OData services.

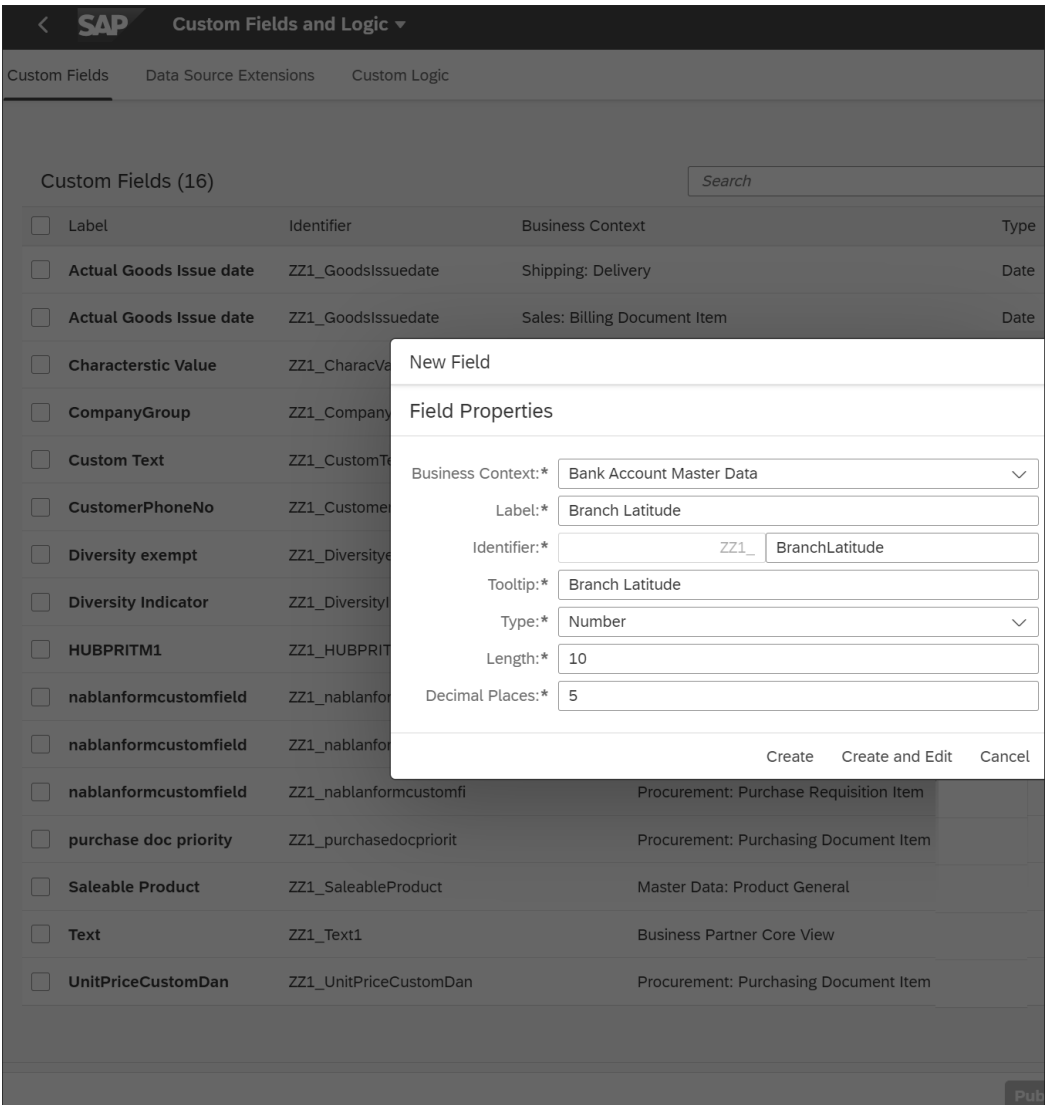


Figure 11.14 Custom Fields App

■ Custom analytics and forms

Figure 11.15 shows the Manage KPIs and Reports app in SAP S/4HANA. Key users can define KPIs using this app. It's your primary app for creating all analytical applications using KPIs, reports, stories, and so on. Created applications can be launched directly from the SAP Fiori launchpad. The following options are available:

- Modify SAP-delivered KPIs.
- Define drilldowns.
- Apply filters.

- Save filtered analyses.
- Define association between KPIs
- Define custom forms and manage output types.

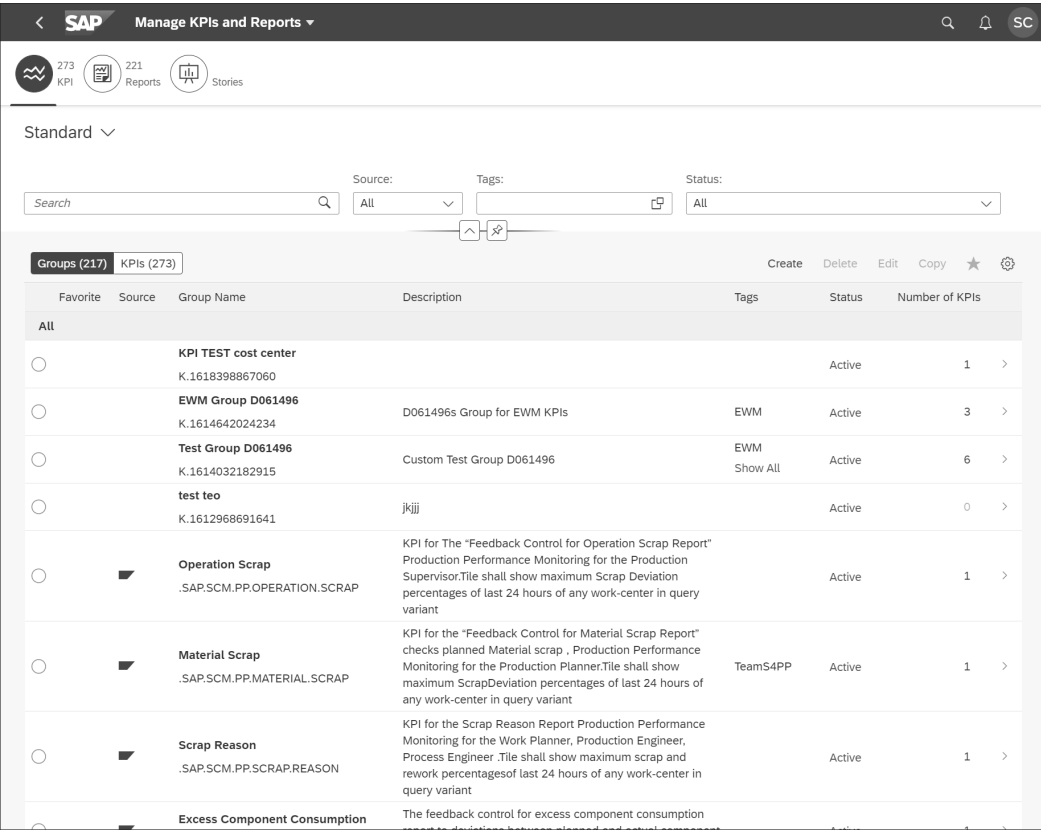


Figure 11.15 Manage KPIs and Reports App

■ Custom business objects

Figure 11.16 shows the Custom Business Object app that gives you the opportunity to create custom business objects in SAP S/4HANA. The Custom Business Object app lets you create your own business objects, add custom fields to existing business objects, create CDS data models, and so on. The following options are available:

- Create your own business objects.
- Add fields to business objects.

- Create CDS views and OData services.
- Define field type associations.
- Create multiple subnodes for business objects.

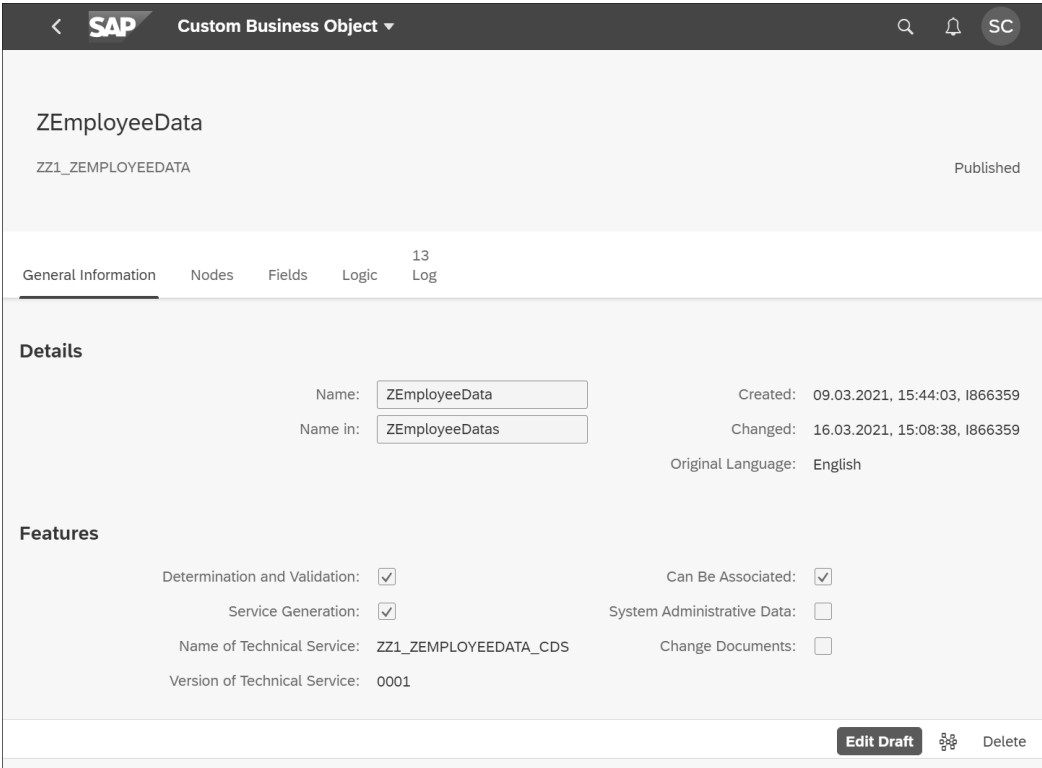


Figure 11.16 Custom Business Object App

■ Custom business logic

Figure 11.17 shows the Custom Fields and Logic app. In this example, you can see how custom business logic is implemented. You can think of this as conceptually the same as business add-ins (BADIs) where you can write custom business logic by plugging in the standard business object behaviors. The following options are available:

- Build your custom logic.
- Easily explore APIs.
- Perform syntax checks.
- Limit to restricted ABAP.

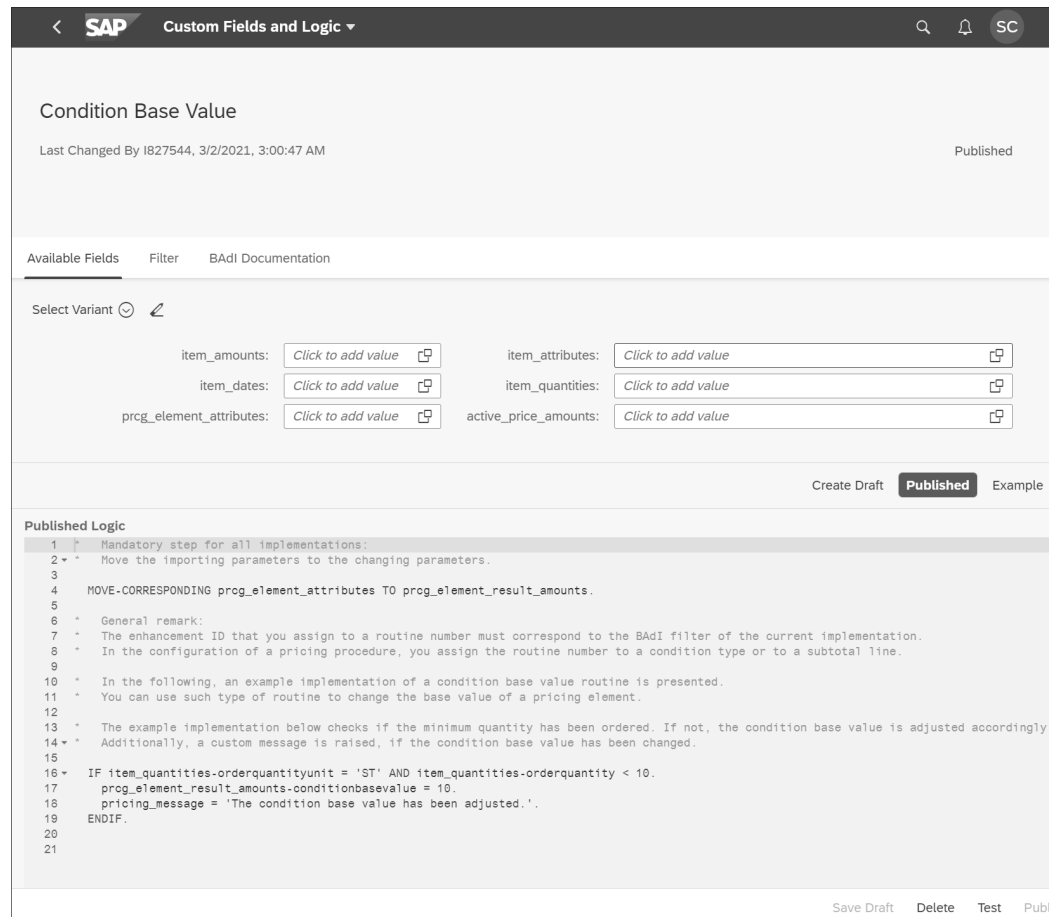


Figure 11.17 Custom Fields and Logic App

Overall, in-app extensions provide a way to infuse critical custom logic within the core SAP S/4HANA functionality. Even though in-app options are fairly limited, using these options is the future-proof way to integrate custom logic within the core functionality. Figure 11.18 depicts how these in-app extension technologies are well integrated with standard SAP functionality, providing powerful extensibility options while keeping the core clean. The shaded parts are SAP standard deliverable objects. The unshaded parts are the custom UI, custom business objects, UI adaptation, and so on that let you make in-app enhancements.

However, there are limitations on how much you can achieve with in-app extensions. When you look at an extension request, you should try to check if this is possible within the in-app extensibility framework. When the extension requirements are fairly simple, you can accommodate this to the in-app framework. However, more complex requirements demand more complex development that can't be realized by power users only. In such cases, developers are required to build applications.

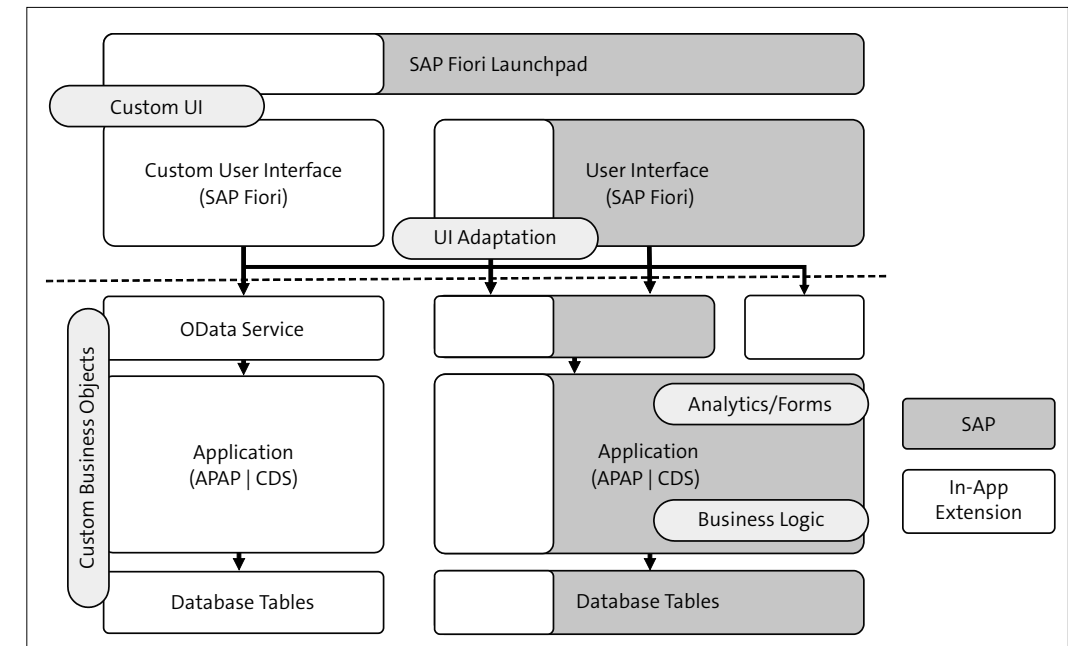


Figure 11.18 In-App Options in SAP S/4HANA

Figure 11.19 shows the limitation of key user extensibility. If the business requirement is to implement more complex logic, then you need to use side-by-side extensibility where developer-intensive work is necessary as opposed to key user extensibility where power business users can manage the extensions. We'll discuss these extensions next.

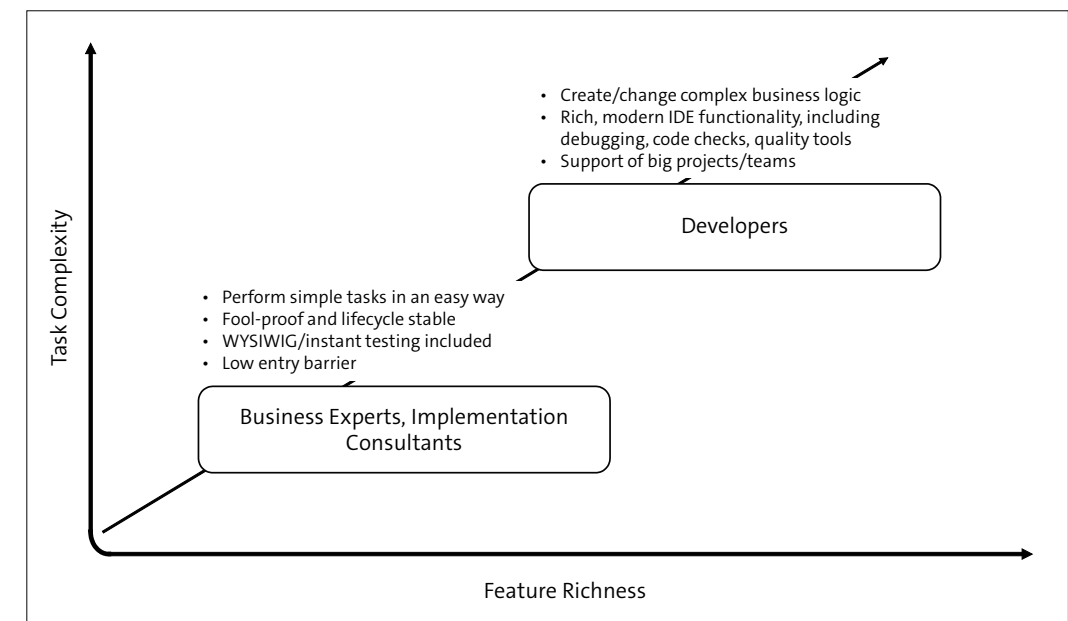


Figure 11.19 Development Trend Based on Feature and Complexity

### 11.3.2 Side-by-Side Extensibility

Side-by-side extensibility covers the bulk of scenarios that can't be accommodated with in-app extensions. This is where SAP BTP comes in handy to implement these loosely coupled extensibility techniques. Side-by-side extensions open up the door of possibilities. Unlike in-app, you have a wide choice of UI technologies runtimes, programming languages, programming models, databases, and architecture styles that can be leveraged in side-by-side extension (see Figure 11.20).

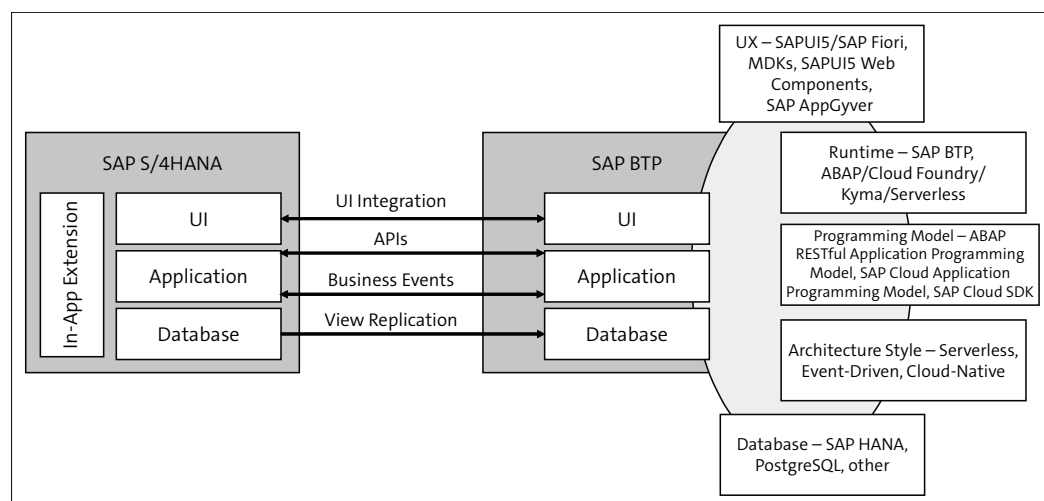


Figure 11.20 Side-by-Side Extensibility Mechanisms

For side-by-side extensibility, different services from SAP Extension Suite are leveraged. Typically, we have to take care of the core data model extension, which can result in custom tables, views, or complex persistent objects such as unstructured data. SAP Extension Suite supports a host of relational and in-memory database solutions for structured data and an ObjectStore for unstructured data that can be flexibly integrated to underlying hyperscaler offerings. We discussed these options in detail in Chapter 7.

While developing the core application, SAP BTP provides you different choices of runtimes. We've already covered details of these runtimes and decision metrics on which one suits which type of extension in Chapter 5.

Similarly, SAP BTP provides flexibility of programming model and language choice. For example, SAP Cloud Application Programming Model can be implemented using Node.js or Java, while the ABAP RESTful application programming model relies on ABAP RESTful programming. In addition, you can build apps using available Cloud Foundry runtimes such as Python or Go. While building cloud-native apps, SAP BTP gives you the choice of having a Kubernetes-based Kyma runtime that is highly scalable and flexible, and you can use virtually any language and runtime using containers. For UI technologies, you also have different choices. You can either follow SAPUI5-based development or use SAP Fiori elements if your design adheres to SAP Fiori principles.

Additionally, you have MDK web version and low code/no code (LC/NC) platforms such as SAP AppGyver to build apps quickly and easily.

### 11.3.3 In-App versus Side-by-Side Extensibility

We often hear the buzzword phrase: keep the core clean. Fundamentally, keeping the core clean means that you shouldn't do extensive modifications to the core standard solution so that your overall solution is jeopardized when upgrading in the future. We'll discuss some more details to help understand this concept.

Figure 11.21 shows the dos and don'ts for keeping the core clean. These are some general recommendations and guidelines that should be followed during an implementation. It shows a comparative review of a traditional extension approach versus a modern extensibility approach where you rely on whitelisted BAdIs, APIs, events, and virtual data models, unlike the traditional approach of core modification, user includes, direct database writes, dynamic calls, implicit enhancement spots, code generation, and so on.

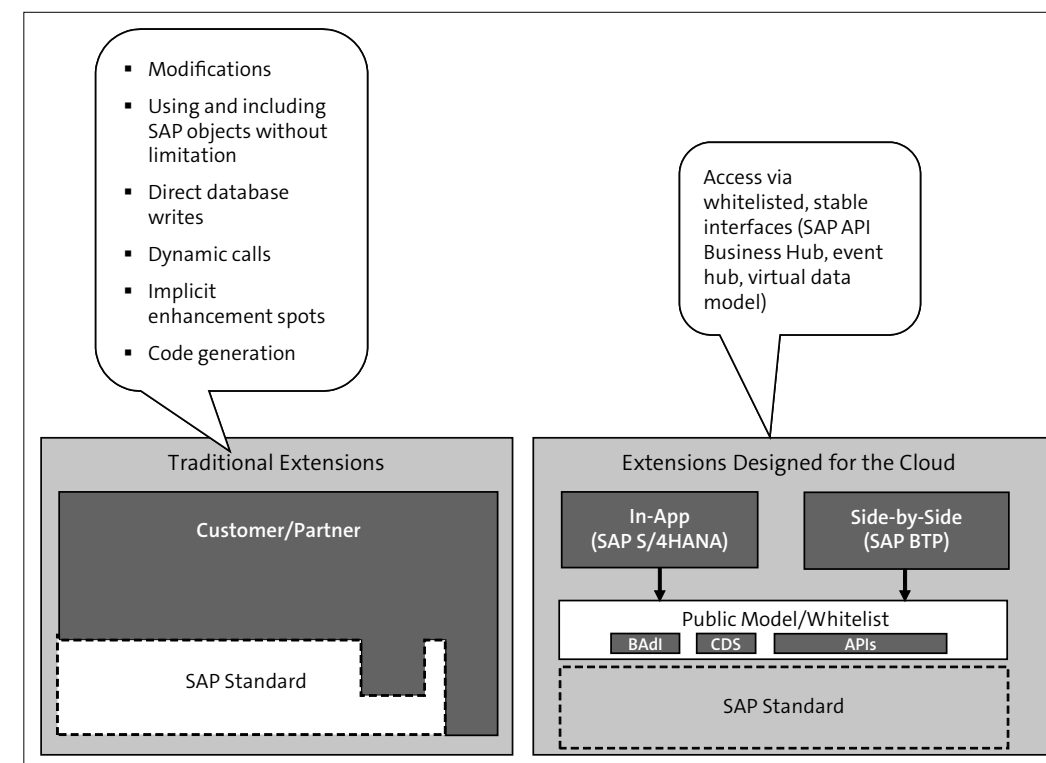


Figure 11.21 Keep the Core Clean: The Right Approach

During implementation cycles, we often encounter questions on what a litmus test is to check whether we're following the right guidelines to keep the core clean. Figure 11.22 provides a clean core index that serves as this litmus test. Applying zero modification



is the goal. However, in most cases, that's not practical. You need to take it case by case and go through these checks step by step.

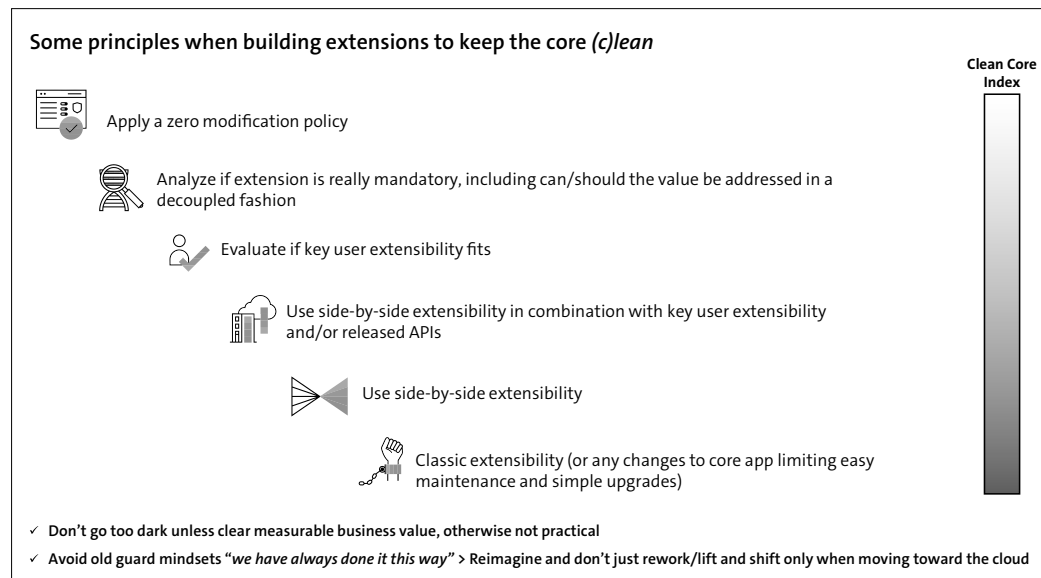


Figure 11.22 Litmus Meter to Keep the Core Clean

One of the key questions that we face commonly is whether to develop an in-app versus side-by-side extension. The previous two sections already described in detail the possibilities with an in-app framework. One rule of thumb is to check first whether the requirement can be met by the in-app limitations. If not, then you need to think of side-by-side extensions. The matrix shown in Figure 11.23 provides a high-level guideline on the scope horizon for in-app extensions and where the horizon is extended for side-by-side extensions. It shows what type of use cases are best fit for in-app extensions and what type of developments are best fit for side-by-side extensions.

SAP S/4HANA provides business configurations that are similar to traditional SAP IMG. On top of this, SAP S/4HANA gives some degree of flexibility for power users in terms of UI adaptation and individual terminology definition. The next level of extensions can be achieved using custom UIs, custom business objects, custom business logic, and so on, which falls under the in-app framework itself. Beyond this, when you need to build a complete custom business process or extend the reach of the app to a wider audience, you need to build a complete custom app. Then, you're gradually entering the domain of side-by-side extensions.

Notice that in certain scenarios, you can even use combinations of these two different techniques to create a hybrid approach. There are many scenarios where you'd need to add an in-app extension point to kick off or integrate a process, but then the majority of the extension is built using the side-by-side technique.

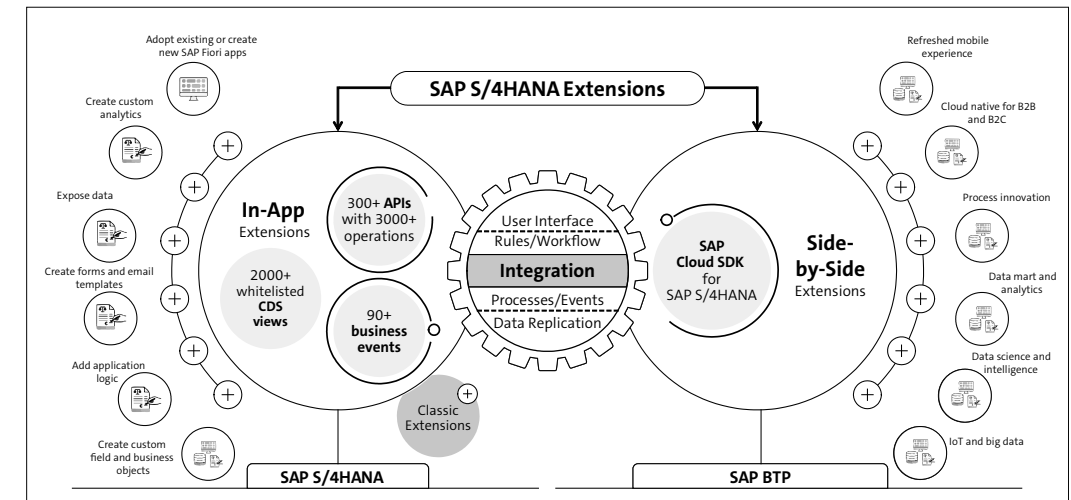


Figure 11.23 In-App versus Side-by-Side Scope Spectrum

Let's consider a few hybrid examples. First, Figure 11.24 shows the technique where both in-app and side-by-side extensibility frameworks are in action. In this example, core OData is built using an in-app extension, and the main extension is built using SAP Cloud Application Programming Model and SAP Cloud SDK side-by-side extensions.

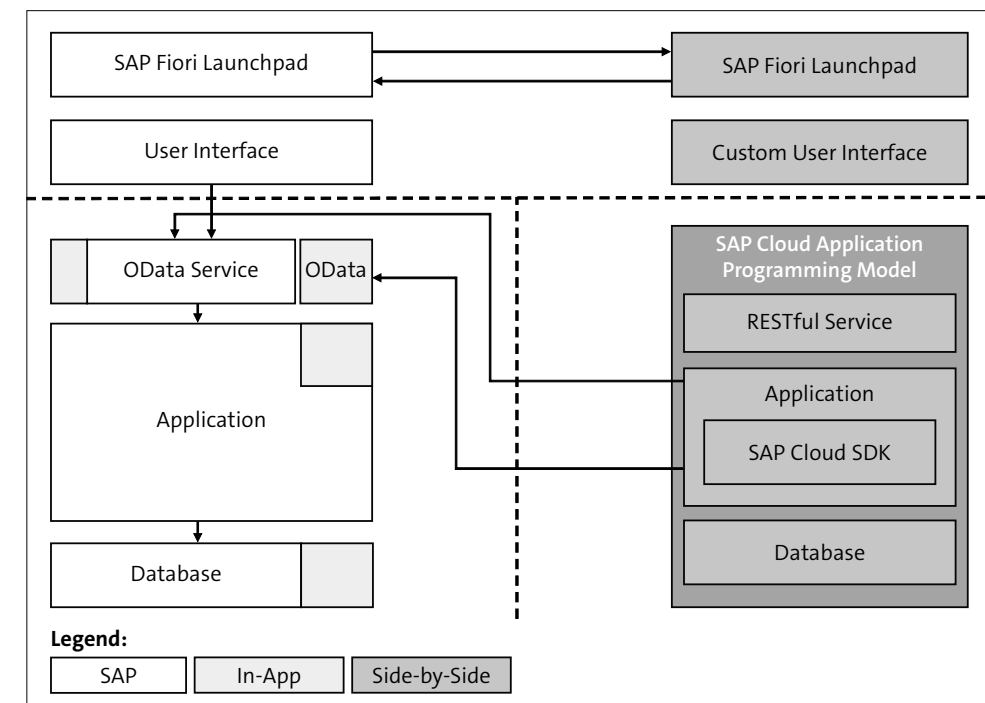


Figure 11.24 Side-by-Side Extension Using SAP Cloud Application Programming Model and Custom SAPUI5

Here we're developing a custom OData service by exposing a custom business object as a service. Then we need to do the necessary configurations for communication arrangements and connectivity setup. Again, we can use custom business logic to drive custom logic for this custom business object. On the SAP BTP side, we must use the destination service to manage the connectivity to SAP S/4HANA. We can either use SAP Cloud Application Programming Model to call this OData via SAP Destination service or combine the SAP Cloud SDK framework to make type safe calls to the SAP S/4HANA backend.

#### Note

Usually, different backends carry different semantics and syntax of business objects. *Type safe* means SAP Cloud SDK provides a virtual abstraction layer to harmonize this syntax so that, as a programmer, you need not handle low-level syntax.

Ultimately, we need to build a service on top of this that will provide the harmonized service layer on top of this data model, comprising custom data entities stored in SAP BTP databases (e.g., SAP HANA) as well as entities that ingest data from the OData services via SAP Cloud SDK. From the UX standpoint, the service will provide access to the entities seamlessly. If you have UX requirements where you need to develop a custom UI far from SAP Fiori design principles, you can develop the UI as a separate app and consume the service exposed from the SAP Cloud Application Programming Model layer. This extension app then can be integrated to the SAP Fiori launchpad.

Figure 11.25 shows another hybrid example where the core OData is built using an in-app extension. Then, a side-by-side approach is taken using the ABAP RESTful application programming model. We can consume remote function call (RFC) or Simple Object Access Protocol (SOAP) services as well using the service consumption model.

As you can see, the side-by-side part can be done using the ABAP RESTful application programming model. Unlike the previous example, here the ABAP RESTful application programming model is providing the service layer. So, SAP BTP provides a flexible framework by which you may choose different programming models and runtimes to realize the service layer.

Similarly, you could use the UX layer by annotation-based SAP Fiori elements or another UI framework.

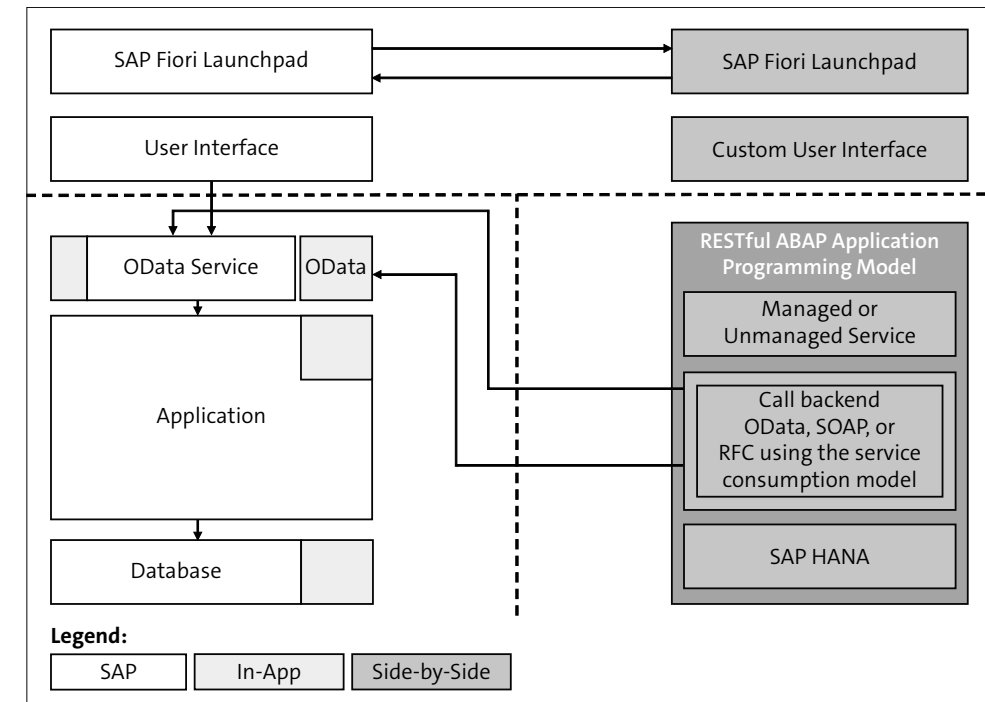


Figure 11.25 Side-by-Side Extensions Using the ABAP RESTful Application Programming Model

#### 11.3.4 Mapping Development Objects

A bulk of SAP S/4HANA implementation projects start from a migration path. For most customers, there might be some version of an ERP core system already existing, and the most common scenario starts with a baseline existing solution. This section addresses, on a high level, the different techniques that can be adapted to identify the right candidates that should be implemented outside the core to keep the core clean and which SAP BTP services can be leveraged to achieve similar functionality using side-by-side extensibility. We'll walk through the various reports, interface, conversion, enhancements, forms, and workflow (RICEFW) objects.

#### Note

In an implementation, side-by-side isn't the default path. You have to judge case by case to see what the right candidates are. There are many scenarios where you should first try to fit the solution to in-app extensibility.

Custom Code Analysis

In a typical project where the existing system already has a lot of custom code, you need to migrate the custom code to the new to-be landscape. One of the pitfalls in migrating as is, without thinking deeply, is the tendency to do the same thing in the same way. SAP S/4HANA isn't only the natural successor of your ERP system; it also comes with new possibilities. And coupled with SAP BTP, you can do even more in SAP S/4HANA to build custom solutions in an upgrade-safe way. So, the first thing you should do is analyze the existing code. Table 11.1 shows a comparative review on the various capabilities of different tools available for custom code analysis.

	Custom Code Migration App	Remote ABAP Test Cockpit (ATC) with SAP GUI	Custom Code Migration App in SAP BTP
Required System	Central ATC check system (SAP S/4HANA release 1809 and up)	Central ATC check system (Basis 7.52)	SAP BTP, ABAP environment
System Location	Customer landscape	Customer landscape	SAP BTP
Remote Connectivity	Through RFC	Through RFC	Through RFC and the cloud connector
Analyze SAP S/4HANA Findings	Yes	Yes	Yes
Simplification Information in ATC Results	Yes	Yes	Yes
Filter Results by Scope and Quick Fix Availability	Yes	No	Yes
Define Custom Code Migration Scope Based on Usage Data	Yes	No	Yes
Remove Custom Code during System Conversion Using Software Update Manager	Yes	No	Yes

Table 11.1 Comparison of Capabilities of Different Options for Custom Code Analysis

Reports

In a traditional SAP system, you might have reports such as list reports, ABAP List Viewer (ALV) reports, and so on. In the SAP S/4HANA world, where SAP Fiori-based UX

is widely accepted, these traditional SAP GUI-based reports are generally outdated. Unlike the traditional requirements of view-only reports, we're now far more capable of delivering more feature-rich functionalities in reports. We can visualize the reports in more intuitive and interactive ways with engaging graphics, interactive charts, and drilldowns. In addition, you can embed analytical intelligence in the reports, which leads to turning insight into action.

Let's say you have a traditional material aging report in combination with open purchase order information so that you can tally the stock aging and corresponding turn-around time for a material procurement. You traditionally may have an ALV report that is capable of sorting data by highest stock aging or filtering by a material group or supplier. However, if you want to see the purchasing and stock trend of a material, you may need to look for other reporting tools such as SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI). Again, if you would like to take some action, such as notify a supplier for a high demand material, you need to go out of this report and use some other tools or transactions to accomplish the same. Now we have wide range of possibilities with different SAP BTP offerings. If you'd like to simply get more insight and analytics, and the report is more drilldown in nature, SAP Analytics Cloud is the right solution. You can certainly add highly interactive and intuitive drilldown charts and even offer some insight to action through SAP Analytics Cloud.

However, if you think SAP Analytics Cloud is a bit farfetched for your scenario, you can build the interactive reports and in-line insight to action with embedded micro charts using SAPUI5 SAP Fiori elements. With the SAP BTP, ABAP environment, you can now build cutting-edge SAP Fiori elements apps with all these capabilities with far less coding using SAP Fiori elements annotations. Of course, if the UX requirements don't fit the SAP Fiori design guidelines, there is an option to customize in a freestyle approach using SAPUI5 development. The primary solution approach can be laid out in the following way:

- **Think of whitelisting**  
Before jumping onto custom development, check out the whitelisted APIs available for the related data object. You can refer to the SAP API Business Hub (<https://api.sap.com/>) for whitelisted APIs.
- **Modularize core data**  
Instead of traditional report programs, try to modularize the custom logic in RFC function modules, or via custom CDS. This way, the code will be reusable, streamlined, and manageable.
- **Expose service**  
Expose the data via OData services. This can be done in either the traditional way of using Transactions SEGW and /IWFND/MAINT\_SERVICE or simply by using OData annotations in your CDS view.

### ■ Provide the UX

Next, use the OData in the UX layer. This layer can be realized using SAP BTP in the following ways:

- If this is just the UX layer, and you have all the data prepared in the SAP S/4HANA layer already, a simple SAPUI5-based application can be developed.
- If this application layer needs more server-side logic, integration, enhancement, data conglomeration, reconciliation, and so on, then either the SAP BTP, ABAP environment can be used with the ABAP RESTful application programming model or SAP Cloud Application Programming Model can be used with open-source Java or the Node.js framework.
- If the report is embedded with a wider analytics story, then broader platforms such as SAP Analytics Cloud can be used where the exposed OData can be used as the data source to provide cutting-edge analytics reports and stories.

### Interface

Interfaces are integral parts of any SAP implementation. Traditionally, you have Application Link Enabling (ALE)/IDoc, RFCs, or proxies to interact with middleware, which eventually enables data to be exposed or ingested, and processes to be integrated across different solutions. SAP BTP provides a host of services that support these traditional interface techniques and provides modern solutions:

### ■ Message-based integration

It's always recommended to use the standard mechanisms such as whitelisted APIs (OData, SOAP), IDocs, and so on to build interfaces. Middleware tools such as SAP Process Integration or the Cloud Integration capability of the SAP Integration Suite can be leveraged to build full-fledged message-based integrations.

### ■ Connectivity

SAP Process Integration/SAP Process Orchestration and the Cloud Integration capability support a wide range of different adapters for different protocols such as HTTP, File, OData, MQ, IDoc, and more. Apart from this, you can now use the Open Connectors capability, which provides out-of-the-box standard connectors for 1,000+ industry-specific connectors. You can find the full list of connectors at <http://s-prs.co/v544032>.

### ■ Event-based integration

In addition to the standard communication scenarios that support APIs, IDocs and so on, a new breed of real-time integration opportunity is now emerging with SAP BTP. SAP is now embedding business events within the core. This means that, if you want to send a notification to an external party in real time at the time of occurrence of a business event, for example, then you can leverage the corresponding business event to trigger this interface. A detailed discussion of event-driven architecture and how SAP Event Mesh helps orchestrate these scenarios was provided in Chapter 10.

### ■ API management

Traditionally, the message broker is the most commonly found integration pattern where middleware does the message mediation, mapping, and routing. Unlike the message broker pattern, the other emerging pattern prevailing in the industry is the API Management capability. Often, we have enterprises with diversified landscapes in need of a common set of API strategies to expose, govern, and monetize functionalities within intracompany or intercompany interactions. API Management comes with full-scale capabilities of API management, throughput control, throttling, implementing security policies, and governing API usages.

### ■ Custom interfaces

There might be requirements to develop custom interfaces, which typically were fulfilled by extending an IDoc or developing a core custom RFC. In a situation in which keeping the core clean is the buzzword, you need to be careful when jumping on such developments. Instead, you should explore other possibilities. For example, if you need to build a data model for these custom interfaces, you might check if this can fit into in-app framework. If not, then you might need to build a side-by-side extension using the ABAP RESTful application programming model in the SAP BTP, ABAP environment, or using SAP Cloud Application Programming Model in SAP HANA Cloud, while letting the middleware handle the bulk of the custom logic. With SAP BTP and additional core capabilities such as the event bus, CDS, and so on, you now have far more different capabilities that can be leveraged to achieve such scenarios.

### Conversion

Typically, conversion is one area where we don't see greater involvement of side-by-side extensions. Usually, SAP products provide inherent tooling for system conversion. For example, SAP S/4HANA comes with a set of system conversion tools such as Silent Data Migration Infrastructure (SDMI), or SAP SuccessFactors provides the Data Upload tool, which provides rich functionality to do data migration and conversion during project implementation. However, in some unusual circumstances, you might find some of the SAP BTP services handy to provide custom solutions to custom conversion requirements:

### ■ Spreadsheet upload

This is a very common scenario in which you need to upload data from a spreadsheet. If this is a simple spreadsheet upload with a limited number of rows, and you need to view and scrutinize the uploaded data before posting, then a SAPUI5-based SAP Fiori app can be built.

### ■ Automation

If the scenario demands more repetitive tasks, such as similar spreadsheets uploaded frequently, and the steps are fairly mechanized, then you may opt for SAP Intelligent RPA to handle these tasks.



## Enhancement

Enhancement is probably the most critical piece where SAP Extension Suite can be leveraged. We discussed in-app and side-by-side extensions in detail in the previous sections. However, one primary mindset in the development approach is to keep the litmus test shown earlier in Figure 11.22. When designing an enhancement, the sequence should be prioritized to ensure that the principle of keeping the core clean isn't breached. Even though in a previous section, we mentioned in-app and side-by-side extensibility techniques, for most practical purposes, we have to adopt a hybrid approach—where necessary development is done close to the core using an in-app extension to ensure integrity of the enhancement to the core solution, but we also use the freedom of the side-by-side extension.

## Forms

In a traditional SAP ERP landscape, you have full freedom to create forms and customize them. In SAP S/4HANA, SAP gives you some in-app capabilities to customize form and output management. Using the Maintain Form Template app in SAP S/4HANA, you can get the list of available form templates, download them, and optionally customize them. You can use Adobe LiveCycle Designer to view the data source and fields or customize forms. These customizations are limited to adding logos, adding footer text, making layout changes, adding custom fields, and so on. However, there might be scenarios where you need deeper customization such as building a form based on a custom data source or merging the output from multiple data sources; in those cases, the only way is to rely on a side-by-side extension.

To develop a side-by-side development for forms, you need the following SAP BTP services:

- **SAP Forms service by Adobe**

This service lets you print and generate interactive forms using Adobe Document Services (ADS). You can call the forms service using REST APIs. Using REST APIs, you can store templates in the service via the template store feature.

- **SAP BTP business rules**

In addition to the core forms services, you can combine complex business rules for certain custom fields using the SAP BTP-based rules service.

- **UX**

Eventually, these custom forms can be called via REST services. However, the entry point of the application could be managed via a custom SAPUI5 app.

## Workflow

Within your enterprise core, many business processes demand workflows. Traditionally, we used to have different functional areas such as sales and distribution, procurement, production planning, and materials management tightly integrated in a single

enterprise core, that is, SAP ERP. In such a tightly integrated environment, many of your end-to-end business processes start and end within the horizon of SAP ERP itself. Even in the SAP S/4HANA world, there might be many business processes that are still close to the core, and the entire horizon of the scope of the process is limited to the boundary of SAP S/4HANA. In such cases, the traditional workflow development is still valid.

However, in this dynamic market where enterprises are competing every day with newer innovations, a more diverse, LoB-specific solution landscape is more common. In such diverse and distributed landscapes, you won't have the end-to-end process confined in a single box as in SAP ERP. Often, we see business processes spanning different LoB solutions. For example, the HR data and organization structure comes from SAP SuccessFactors, the core finance is in SAP S/4HANA, and procurement is driven out of SAP Ariba. This diversity might include other LoB solutions from non-SAP vendors as well. In such scenarios, you need to think of cross-product workflows (refer to Section 11.1.5). We're seeing immense benefits in providing such cross-product workflows that not only integrate different solutions via a process chain but also keep vital business statistics that provide the nervous system of your business.

SAP Workflow Management is the solution here. Often, we see project teams that have limited vision confined within their own work area. This might cause a silo where individual LoB solutions might build redundant heterogeneous workflow solutions. While many of the SAP and non-SAP LoB solutions provide workflow solutions within their scope, one thing is necessary to keep in mind: these LoB-specific workflows can't build the bridge between different LoB solutions. So, one idea to nurture is to step back and look at the bigger picture. During the scoping exercise, if you try to find the bigger process chain, you'll often find a better way to solve business problems in a more integrated way. SAP Workflow Management not only provides a lean flexible workflow engine but also a business rules service where business users can define and maintain business rules. Additionally, SAP Workflow Management provides an easily configurable process visibility dashboard that shows the KPI overviews, workflow status, stages, and drilldown reports. This dashboard is the main cockpit from where you can get the overview of the entire cross-product workflow scenarios.

In addition, the My Inbox app is integrated into the SAP BTP launchpad and SAP Work Zone, which provides a single stop to manage all your workflow tasks. With the modern SAPUI5 and forms capabilities, My Inbox app provides a flexible, intuitive, and appealing UX for your task management.

## 11.4 Summary

In this chapter, we explored how SAP BTP can be leveraged in different business scenarios. We aimed to describe the scenarios in a more generic way so that you can relate

them to your scenarios and finding matching patterns. Though the application of SAP BTP is no way limited to these patterns only, these are certainly some of the most common scenarios. Hopefully this will give leaders like Allan from Rocking Gas good ideas on use case patterns, guiding principles, and some of the core value propositions of SAP BTP in their respective business scenarios.

In the next chapters, we'll dive into more specific business problems and see how you can leverage SAP BTP to solve them. We'll start with how to manage multiple SAP systems via extensions and integrations in the next chapter.

# Contents

Preface .....	17
---------------	----

## PART I Getting Started

### 1 Introduction 23

---

<b>1.1 Digital Transformation and Intelligent Enterprises .....</b>	<b>23</b>
1.1.1 Digital Natives and Digital Immigrants .....	24
1.1.2 Digital Transformation .....	25
1.1.3 Intelligent Enterprise .....	27
<b>1.2 The Role of SAP Business Technology Platform .....</b>	<b>32</b>
<b>1.3 Technology Pillars .....</b>	<b>39</b>
1.3.1 Database and Data Management .....	40
1.3.2 Analytics .....	49
1.3.3 Application Development and Integration .....	51
1.3.4 Intelligent Technologies .....	53
<b>1.4 Summary .....</b>	<b>55</b>

### 2 Account Setup and Administration 57

---

<b>2.1 Trial and Commercial Models .....</b>	<b>57</b>
<b>2.2 Services .....</b>	<b>61</b>
2.2.1 Services Discovery .....	61
2.2.2 Estimator Tool .....	66
<b>2.3 Creating Your Account .....</b>	<b>67</b>
<b>2.4 Account Structure and Setup .....</b>	<b>72</b>
2.4.1 Account Structure .....	72
2.4.2 Setup .....	75
<b>2.5 Administration .....</b>	<b>76</b>
2.5.1 Global Account .....	76
2.5.2 Subaccounts .....	80
2.5.3 Entitlements .....	80
2.5.4 Boosters .....	84
<b>2.6 Summary .....</b>	<b>85</b>

## PART II Application Architecture and Design

### 3 Architecture Design Patterns and Development Approach

89

<b>3.1 Lean Enterprise Architecture</b>	90
3.1.1 Architecture Development Method	90
3.1.2 Design Thinking and Architectural Thinking	92
3.1.3 Lean Enterprise Architecture Toolkit	94
<b>3.2 Architecture Design Patterns</b>	107
3.2.1 Microservice-Based Architecture	107
3.2.2 Cloud-Native Architecture	114
3.2.3 Serverless Architecture	117
3.2.4 Event-Driven Architecture	118
<b>3.3 SAP BTP Runtimes to Support Architecture Patterns</b>	121
3.3.1 Cloud Foundry	121
3.3.2 Kyma	124
<b>3.4 Development Approach</b>	127
3.4.1 Connect	128
3.4.2 Build	131
3.4.3 Add Channels	134
3.4.4 Deploy	135
3.4.5 Operate	135
<b>3.5 SAP Application Extension Methodology</b>	136
<b>3.6 Summary</b>	138

### 4 User Interface

141

<b>4.1 Overview</b>	141
<b>4.2 User Interface Access Channel</b>	142
4.2.1 SAP Launchpad Service	142
4.2.2 SAP Work Zone	151
4.2.3 Comparative Review	156
<b>4.3 App Router</b>	160
<b>4.4 User Interface Development Tools and Technologies</b>	161
4.4.1 SAP Fiori	161
4.4.2 SAPUI5 and OpenUI5	163

4.4.3 Web Components	164
4.4.4 SAP Fiori Elements and SAP Fiori Tools	164
4.4.5 SAP BTP SDK for iOS and Android	169
4.4.6 Mobile Development Kits	170
4.4.7 SAP Screen Personas	172
4.4.8 SAP AppGyver	172
<b>4.5 Summary</b>	173

### 5 Business Logic

175

<b>5.1 Overview</b>	175
<b>5.2 Runtimes</b>	176
5.2.1 SAP BTP, Cloud Foundry Runtime	176
5.2.2 SAP BTP, Serverless Runtime	179
5.2.3 SAP BTP, Kyma Runtime	180
5.2.4 SAP BTP, ABAP Environment	182
5.2.5 Comparative Review	183
<b>5.3 Programming Models</b>	185
5.3.1 SAP Cloud Application Programming Model	185
5.3.2 ABAP RESTful Application Programming Model	187
5.3.3 Comparative Review	190
<b>5.4 SAP Workflow Management</b>	191
5.4.1 Business Rules	193
5.4.2 Workflow	205
5.4.3 Process Visibility	207
5.4.4 Process Flexibility	209
<b>5.5 Summary</b>	211

### 6 System and Process Integration

213

<b>6.1 Overview</b>	213
<b>6.2 SAP Integration Solution Advisory Methodology</b>	214
6.2.1 Assess Your Integration Strategy	216
6.2.2 Design Your Hybrid Integration Platform	223
6.2.3 Define Integration Best Practices	228
6.2.4 Enable a Practice of Empowerment	234



<b>6.3</b>	<b>API Management</b>	239
<b>6.4</b>	<b>Open Connectors</b>	243
<b>6.5</b>	<b>Integration Advisor</b>	247
<b>6.6</b>	<b>SAP Event Mesh</b>	251
<b>6.7</b>	<b>SAP Graph</b>	255
<b>6.8</b>	<b>Destination and Connectivity Service</b>	257
<b>6.9</b>	<b>Process Integration and Orchestration</b>	261
<b>6.10</b>	<b>Summary</b>	264

## **7 Database and Data Integration** 265

<b>7.1</b>	<b>Overview</b>	265
<b>7.2</b>	<b>Data Persistency</b>	266
7.2.1	SAP HANA and SAP HANA Cloud	266
7.2.2	SAP Adaptive Server Enterprise	273
7.2.3	PostgreSQL	274
7.2.4	Unstructured Data	279
<b>7.3</b>	<b>Data Integration</b>	280
7.3.1	SAP Data Intelligence	280
7.3.2	Smart Data Integration	286
7.3.3	SAP Cloud Integration for Data Services	289
7.3.4	On-Premise Integration Tools	292
<b>7.4</b>	<b>Comparative Review</b>	294
<b>7.5</b>	<b>Summary</b>	294

## **8 Analytics** 295

<b>8.1</b>	<b>Overview</b>	295
<b>8.2</b>	<b>SAP Analytics Cloud</b>	299
8.2.1	Analytics Capabilities	300
8.2.2	User Interfaces	307
8.2.3	Platform	308
<b>8.3</b>	<b>SAP Data Warehouse Cloud</b>	312

<b>8.4</b>	<b>Comparative Review</b>	318
8.4.1	SAP Analytics Cloud and SAP BPC	318
8.4.2	SAP Analytics Cloud and SAP BusinessObjects BI	319
8.4.3	SAP Analytics Cloud and SAP IBP	319
8.4.4	SAP Analytics Cloud and SAP Data Warehouse Cloud	320
8.4.5	SAP Data Warehouse Cloud and SAP BW/4HANA	321
<b>8.5</b>	<b>Summary</b>	322

## **9 Security** 323

<b>9.1</b>	<b>Overview</b>	323
<b>9.2</b>	<b>Authentication</b>	324
<b>9.3</b>	<b>Common Security Flows</b>	329
9.3.1	SAML	329
9.3.2	OAuth	330
9.3.3	OpenID Connect	335
9.3.4	Authentication Flows Supported by SAP BTP	336
<b>9.4</b>	<b>Authorization</b>	338
9.4.1	User Account and Authentication Service	339
9.4.2	JSON Web Token	339
9.4.3	App Router and Role-Based Access	341
9.4.4	Authorization Using SAP BTP Cockpit	343
<b>9.5</b>	<b>Cloud Connector</b>	344
9.5.1	Security Considerations	345
9.5.2	Connecting Kubernetes to On-Premise Systems	347
<b>9.6</b>	<b>Identity Propagation</b>	350
9.6.1	OAuth 2.0 SAML Bearer Assertion	351
9.6.2	Principal Propagation	352
<b>9.7</b>	<b>Summary</b>	354

## **10 DevOps** 355

<b>10.1</b>	<b>Overview</b>	355
<b>10.2</b>	<b>Plan and Set Up</b>	356
10.2.1	SAP Continuous Integration and Delivery	357
10.2.2	Project “Piper”	359

10.2.3	Continuous Integration and Delivery Best Practices Guide .....	361
10.2.4	Compatability .....	362
<b>10.3</b>	<b>Develop and Test .....</b>	<b>363</b>
<b>10.4</b>	<b>Deliver and Change .....</b>	<b>366</b>
<b>10.5</b>	<b>Monitor and Operate .....</b>	<b>373</b>
<b>10.6</b>	<b>Implementation .....</b>	<b>378</b>
<b>10.7</b>	<b>Summary .....</b>	<b>380</b>

## PART III Use Cases

### 11 Identifying Use Cases 383

<b>11.1</b>	<b>Types of Innovation .....</b>	<b>383</b>
11.1.1	Extending Functionality .....	384
11.1.2	Extending Reach .....	387
11.1.3	Process Innovation .....	389
11.1.4	Process Automation .....	390
11.1.5	Cross-Product Workflow .....	392
11.1.6	Cross-Functional Business Processes .....	395
11.1.7	Machine Learning .....	396
11.1.8	Internet of Things .....	399
<b>11.2</b>	<b>Identify Business Gaps .....</b>	<b>400</b>
<b>11.3</b>	<b>Extensibility Concept .....</b>	<b>403</b>
11.3.1	In-App Extensions .....	403
11.3.2	Side-by-Side Extensibility .....	410
11.3.3	In-App versus Side-by-Side Extensibility .....	411
11.3.4	Mapping Development Objects .....	415
<b>11.4</b>	<b>Summary .....</b>	<b>421</b>

### 12 Managing Multiple ERP Systems by Extending and Integrating SAP S/4HANA 423

<b>12.1</b>	<b>Description .....</b>	<b>424</b>
<b>12.2</b>	<b>Architecture .....</b>	<b>426</b>

<b>12.3</b>	<b>Selecting SAP BTP Services .....</b>	<b>431</b>
<b>12.4</b>	<b>Implementation .....</b>	<b>433</b>
12.4.1	Connect .....	433
12.4.2	Build .....	436
12.4.3	Add Channel .....	438
12.4.4	Deploy and Operate .....	439
<b>12.5</b>	<b>Summary .....</b>	<b>441</b>

### 13 Managing Procurement Spend by Extending SAP Ariba 443

<b>13.1</b>	<b>Description .....</b>	<b>444</b>
<b>13.2</b>	<b>Architecture .....</b>	<b>446</b>
<b>13.3</b>	<b>Selecting SAP BTP Services .....</b>	<b>451</b>
<b>13.4</b>	<b>Implementation .....</b>	<b>453</b>
13.4.1	Connect .....	453
13.4.2	Build .....	454
13.4.3	Add Channel .....	455
13.4.4	Deploy and Operate .....	456
<b>13.5</b>	<b>Summary .....</b>	<b>457</b>

### 14 Curating a Guided Prehire Experience by Extending SAP SuccessFactors 459

<b>14.1</b>	<b>Description .....</b>	<b>459</b>
<b>14.2</b>	<b>Architecture .....</b>	<b>462</b>
<b>14.3</b>	<b>Selecting SAP BTP Services .....</b>	<b>466</b>
<b>14.4</b>	<b>Implementation .....</b>	<b>467</b>
14.4.1	Connect .....	467
14.4.2	Build .....	467
14.4.3	Add Channel .....	469
14.4.4	Deploy and Operate .....	470
<b>14.5</b>	<b>Summary .....</b>	<b>475</b>

<b>15 Integrating the COVID-19 Risk Index with SAP Concur</b>	477
<b>15.1 Description</b>	478
<b>15.2 Architecture</b>	480
<b>15.3 Selecting SAP BTP Services</b>	484
<b>15.4 Implementation</b>	485
15.4.1 Connect	485
15.4.2 Build	486
15.4.3 Add Channel	489
15.4.4 Deploy and Operate	490
<b>15.5 Summary</b>	491
<b>16 Integrating SAP Customer Experience with Qualtrics</b>	493
<b>16.1 Description</b>	494
<b>16.2 Architecture</b>	496
<b>16.3 Selecting SAP BTP Services</b>	500
<b>16.4 Implementation</b>	501
16.4.1 Connect	502
16.4.2 Build	506
16.4.3 Add Channel	507
16.4.4 Deploy and Operate	509
<b>16.5 Summary</b>	511
<b>17 Building Multitenant Applications Using SAP Cloud Application Programming Model</b>	513
<b>17.1 Description</b>	514
<b>17.2 Architecture</b>	514
<b>17.3 Selecting SAP BTP Services</b>	516

17.3.1 SAP Cloud Application Programming Model	516
17.3.2 SAP BTP, ABAP Environment	519
<b>17.4 Implementation</b>	520
17.4.1 SAP Cloud Application Programming Model	520
17.4.2 SAP BTP, ABAP Environment	530
<b>17.5 Summary</b>	532
<b>18 Building Cloud-Native Applications Using SAP BTP, Kyma Runtime</b>	533
<b>18.1 Description</b>	534
<b>18.2 Architecture</b>	536
<b>18.3 Selecting SAP BTP Services</b>	538
<b>18.4 Implementation</b>	539
18.4.1 Initial Setup and Dependencies	539
18.4.2 Deployment	542
18.4.3 SAP Launchpad Service	545
18.4.4 API Connectivity	547
18.4.5 Scaling	550
<b>18.5 Summary</b>	553
The Authors	555
Index	557

# Index

## A

ABAP .....	108, 182, 188, 487, 519, 530
<i>multitenant applications</i> .....	519
<i>pipeline</i> .....	359, 366
<i>unit test</i> .....	365
ABAP Development Tools (ADT) .....	133, 182, 198, 485
ABAP List Viewer (ALV) report .....	416
ABAP-Managed Database Procedures (AMDP) .....	530
ABAP programming model for SAP Fiori ....	187
ABAP RESTful application programming model .....	113, 133, 175, 187–188, 190, 198, 410, 414, 489, 519, 530
<i>artifacts</i> .....	189
ABAP Test Cockpit (ATC) .....	365, 416
Access channel .....	142
Access rights .....	342
Access token .....	197, 331, 334–335
Account setup .....	75
Account structure .....	72, 74
ActualLRP .....	123
Add channel phase .....	134, 438, 455, 469, 489, 507
Administration .....	76
Adobe LiveCycle Designer .....	420
Advanced Message Queuing Protocol (AMQP) .....	179
Aggregator .....	233
Agile development .....	115
Alert notification service .....	374, 379
Algorithm .....	396, 454
<i>categories</i> .....	397
Analytical application .....	307
Analytics .....	34, 49, 295–296
<i>catalog</i> .....	308
<i>comparative review</i> .....	318
<i>integration</i> .....	217–218, 223
<i>IoT</i> .....	400
Android .....	169
Apache Eclipse .....	133
API designer .....	242
API gateway .....	242
API Management .....	53, 213, 239, 241, 419
<i>use cases</i> .....	241
API portal .....	242
API server .....	125
Application connector .....	126, 181
Application developer .....	235
Application development and integration .....	35, 51, 131, 267
Application Function Library (AFL) .....	271
Application job .....	488
Application-layer security .....	345
Application lifecycle management (ALM) ....	39, 370, 376
Application programming interface (API) ...	52, 129, 180, 192, 213, 240, 243, 254
<i>business rules</i> .....	196, 201, 438
<i>Cloud Foundry</i> .....	527
<i>connectivity</i> .....	547
<i>destination</i> .....	261
<i>endpoints</i> .....	351, 547
<i>gateway</i> .....	116
<i>lifecycle</i> .....	242
<i>maintain destination</i> .....	486
<i>manage</i> .....	241, 419
<i>OAuth 2.0</i> .....	200
<i>REST</i> .....	487
<i>rules</i> .....	547
<i>SAP Analytics Cloud</i> .....	311
<i>SAP Graph</i> .....	256
<i>security</i> .....	326, 330
<i>transport nodes</i> .....	369
<i>use cases</i> .....	241
Application-to-application (A2A) .....	220
App router .....	160, 341, 439, 517
<i>define</i> .....	341
<i>setup</i> .....	527
<i>types</i> .....	160
Architectural decisions .....	103
Architectural thinking .....	93
Architecture .....	89
<i>blueprint</i> .....	230
<i>building blocks</i> .....	102
<i>cloud-native</i> .....	114
<i>design patterns</i> .....	107
<i>development approach</i> .....	127
<i>event-driven</i> .....	118, 253
<i>microservices</i> .....	107, 110
<i>monolithic</i> .....	108
<i>principles</i> .....	97
<i>roadmap</i> .....	106
<i>runtimes</i> .....	121
<i>serverless</i> .....	117

- Architecture (Cont.)  
  *use case* ..... 426, 446  
  *vision* ..... 90  
  *work products* ..... 95  
Architecture development method  
  (ADM) ..... 90, 93  
Artificial intelligence (AI) ..... 53, 396  
ASC X12 ..... 250  
Asynchronous communication ..... 254  
Asynchronous JavaScript and XML  
  (AJAX) ..... 332  
Attended mode ..... 391  
Auditing ..... 310  
Audit trail ..... 207  
Augmented analytics ..... 298, 303  
  *automated* ..... 303  
  *predictive* ..... 304  
Authentication ..... 122, 324, 350  
  *flows* ..... 329, 336  
Authorization ..... 338, 343  
  *access rights* ..... 342  
  *flows* ..... 330  
  *server* ..... 330, 335, 339  
Automated Predictive Library (APL) ..... 268  
AutoML ..... 454  
Auto-scaling ..... 113, 537
- B**
- Backing service ..... 112  
Baseline solution architecture .... 101, 426, 446,  
  462, 480, 496  
Bimodal IT ..... 26  
Blob store ..... 123, 178  
Boosters ..... 84, 467, 485, 489  
Bootstrap ..... 525  
Bot ..... 392, 454, 456  
Brand experience ..... 31  
Build, release, run ..... 112  
Build phase ..... 131, 363, 436, 454, 486, 506  
Business architecture ..... 91  
Business catalog ..... 316  
Business data graph ..... 256  
Business driver ..... 92  
Business Function Library (BFL) ..... 271  
Business gaps ..... 400  
Business intelligence (BI) ..... 49, 297, 300  
Business layer ..... 316  
Business logic ..... 175, 407  
Business network ..... 32  
Business object ..... 406  
Business Object Processing Framework  
  (BOPF) ..... 188  
Business object projection ..... 189  
Business operational monitoring ..... 234  
Business process monitoring ..... 377  
Business reach ..... 387  
Business rule ..... 193, 394, 420, 431, 436–437,  
  466, 482, 484, 489  
  *API* ..... 196–197, 201, 438  
  *attributes* ..... 194  
  *call in ABAP environment* ..... 198  
  *call in Node.js* ..... 197  
  *call in SAP S/4HANA* ..... 200  
  *consume* ..... 196, 201  
  *text rules* ..... 468  
Business Rule Framework plus (BRFplus) ... 194  
Business services provisioning ..... 189  
Business-to-business (B2B) ..... 222  
Business-to-government (B2G) ..... 222  
Business workflow ..... 208
- C**
- Calculation view ..... 271  
Cardinality ..... 273, 506  
Change and Transport System  
  (CTS) ..... 366, 368, 372  
Change management ..... 366, 369  
Change request management  
  (ChARM) ..... 368–369  
Channel ..... 134, 388  
Chatbot ..... 155, 392, 439  
Circuit breaker ..... 116  
Citizen integrator ..... 235  
Classification ..... 304  
Clean core index ..... 411  
Client credential flow ..... 333  
Client profile ..... 201  
Climate 21 program ..... 32  
Cloud connector ... 144, 225, 344, 352, 431, 434,  
  451, 484, 486  
  *security considerations* ..... 345  
  *shadow instance* ..... 346  
  *technical prerequisites* ..... 346  
Cloud controller ..... 123, 125, 178  
CloudEvents ..... 120, 130  
  *trigger* ..... 179  
Cloud Foundry ..... 73–74, 112, 121, 123, 160,  
  176, 183, 288, 336, 344, 350, 367, 456, 534  
  *APIs* ..... 527  
  *CLI* ..... 178  
Cloud infrastructure layer ..... 347

- Cloud Integration ..... 242, 262, 390, 396, 418,  
  428, 431, 435, 501, 503–504  
Cloud Integration Connector ..... 501  
Cloud management tool ..... 73  
Cloud-native application ..... 533  
Cloud-native architecture ..... 114  
  *characteristics* ..... 115  
  *container-based* ..... 116  
  *microservice-based* ..... 116  
Cloud-Native Computing Foundation  
  (CNCF) ..... 115  
Cloud Platform Enterprise Agreement  
  (CPEA) ..... 59  
Cloud security ..... 324  
Clustering algorithm ..... 454  
Cluster node ..... 126  
Codebase ..... 111  
Code verifier ..... 332  
Collaboration ..... 311, 316  
  *platform* ..... 388  
Collaborative enterprise planning ..... 306  
Column discriminator ..... 514  
Command-line interface (CLI) ..... 136  
  *Cloud Foundry* ..... 178, 206  
Commercial model ..... 57  
  *types* ..... 59  
Common data model ..... 145  
Communication arrangement ..... 189, 198  
Communication Arrangement app ..... 189  
Communication pattern ..... 253  
Communication scenario ..... 189  
Communication system ..... 189  
Communication System app ..... 189  
Communication user ..... 189  
Composed message processor ..... 233  
Conceptual data ..... 104  
Concurrency ..... 113  
Connection Management app ..... 283  
Connection proxy ..... 257  
Connectivity ..... 418  
  *isolation* ..... 515, 518  
  *proxy* ..... 347–348  
Connect phase ..... 128, 433, 453, 467, 485, 502  
Content-based routing ..... 233  
Content enricher ..... 234  
Content explorer ..... 149  
Content federation ..... 145  
Content filter ..... 234  
Content manager ..... 146, 148–149  
Content package ..... 155  
Content provider ..... 546  
Content translation ..... 311  
Continuous integration ..... 361, 367  
Continuous Integration & Delivery app .... 358  
Continuous integration and continuous  
  delivery (CI/CD) ..... 75, 116, 355, 543  
  *compatability* ..... 362  
  *solutions* ..... 357  
Continuous Integration and Delivery Best  
  Practices Guide ..... 357, 361  
Continuous learning ..... 115  
Control plane ..... 125  
Controls ..... 163  
Conversational AI ..... 134  
Conversion ..... 419  
Core data model extension ..... 410  
Core data services (CDS) ..... 185, 187–188,  
  272, 312  
Cross-functional business process ..... 395  
Cross-origin resource sharing (CORS) ..... 332,  
  508  
Cross-product workflow ..... 392  
Cross-use case ..... 218, 220  
Custom Business Object app ..... 406  
Custom code analysis ..... 416  
Custom Code Migration app ..... 416  
Customer experience ..... 30  
Customer feedback ..... 494, 496  
Custom Fields and Logic app ..... 404, 407  
Custom interface ..... 419
- D**
- DaemonSet ..... 126  
Dashboard ..... 448, 474  
Database ..... 40  
Database and data management .... 33, 40, 265  
  *comparative review* ..... 294  
Database instance ..... 515  
Database table ..... 530  
Data catalog ..... 285  
Data center ..... 62, 66  
Data control language (DCL) ..... 272  
Data event ..... 252  
Data exploration ..... 300  
Data flow ..... 292, 314  
Data integration ..... 217–219, 280, 282  
  *patterns* ..... 223  
Data isolation ..... 514, 517  
Data lake ..... 268, 312  
  *IoT* ..... 400  
Data layer ..... 316  
Data management ..... 42  
  *extension solutions* ..... 49



Data modeling ..... 310, 316  
    *harmonization* ..... 245  
Data persistency ..... 266  
    *structured data* ..... 266  
    *unstructured data* ..... 279  
Data preparation ..... 310  
Data privacy ..... 268  
Data processing ..... 284  
Data provisioning ..... 286  
    *agent* ..... 286  
Data quality ..... 297  
Data replication ..... 268, 289, 314  
Data services agent ..... 289  
Data silo ..... 295, 312  
Data store ..... 291  
Data-to-value ..... 38  
Data warehousing ..... 49, 297, 321  
Debugger tool ..... 340  
Decision table ..... 225  
Deliver phase ..... 94  
Demilitarized zone (DMZ) ..... 346  
Dependencies ..... 111  
Deploy and operate phase ... 135, 439, 456, 470, 490, 509  
Design pattern ..... 107  
Design phase ..... 94  
Design thinking ..... 93–94, 99  
DesiredLRP ..... 123  
Desktop agent ..... 452  
Desktop frontend ..... 439  
Destination ..... 257, 336, 345, 451, 484, 502, 549  
    *call API* ..... 261  
    *consume* ..... 260  
    *create* ..... 258  
    *manage* ..... 258  
    *REST APIs* ..... 258  
Developer onboarding ..... 485  
Developer portal ..... 242  
Development ..... 75  
Development/production parity ..... 113  
Development approach ..... 127  
Development efficiency ..... 52  
Development object ..... 415  
DevOps ..... 241, 355, 362, 402  
    *deliver and change* ..... 366  
    *develop and test* ..... 363  
    *implementation* ..... 378  
    *monitor and operate* ..... 373  
    *phases* ..... 356  
    *plan and set up* ..... 356  
Diego ..... 177  
Diego Brain ..... 123  
Diego Bulletin Board System (BBS) ..... 123, 178  
Diego Cell ..... 123, 177  
Digital experience ..... 51  
Digital immigrant ..... 23–24  
Digitalization ..... 25  
Digital native ..... 24  
Digital processes automation ..... 52  
Digital transformation ..... 23, 25  
Digital twin ..... 400  
Digitization ..... 23, 25  
Direct access ..... 144  
Directory ..... 73  
Discover phase ..... 94  
Disposability ..... 113  
Distributed monolithic architecture ..... 109  
Docker ..... 124, 126, 359  
    *image* ..... 178, 359  
Domain model ..... 395  
Dynamic routing ..... 527  
Dynatrace ..... 373

## E

Edge protocol ..... 400  
Element ..... 194  
Email ..... 488  
Employee engagement ..... 461, 463–464, 470  
Enhanced Change and Transport System (CTS+) ..... 366, 368  
Enhancement ..... 420  
Enterprise microservice ..... 241  
Enterprise planning ..... 50, 297, 306  
Enterprise reporting ..... 301  
Enterprise trial ..... 58, 60  
Entitlement ..... 73, 80, 357, 539  
    *assign* ..... 81  
    *set* ..... 73  
    *view* ..... 80, 83  
Entity relationship modeling ..... 104  
Environment and location diagram ... 106, 430, 450, 465, 483, 498  
Estimator tool ..... 66  
Event ..... 129, 205, 251, 254  
    *enablement* ..... 434  
    *types* ..... 251  
Event-based integration ..... 418  
Event broker ..... 254  
Event bus ..... 120, 130  
Event-driven architecture ..... 118, 253  
Exception management ..... 377  
Experience data (X data) ..... 28, 496  
Experience management ..... 30

Explore phase ..... 94  
Expression language (EL) ..... 272  
Extended application services (XS)  
    runtime ..... 269  
Extended planning and analysis (xP&A) ... 298  
Extended services for user account and  
    authentication (XSUAA) service .... 122, 260, 336, 339, 341, 488, 517, 528, 541  
Extensibility ..... 39, 51, 129, 136, 378, 402–403, 431  
    *comparative review* ..... 411  
    *hybrid approach* ..... 412  
    *in-app extensions* ..... 403  
    *innovations* ..... 384  
    *pipelines* ..... 361  
    *SAP Work Zone* ..... 155  
    *side-by-side extensions* ..... 410  
    *use cases* ..... 386  
Extension application pattern ..... 137  
Extension Center ..... 180  
Extension domain ..... 136  
Extension request ..... 408  
Extension task ..... 136  
External user ..... 388  
External workspace ..... 155

## F

Feature set A ..... 73  
Feature set B ..... 73, 257  
Federated identity management ..... 325  
Feedback channel ..... 496, 502  
Feeds ..... 152  
Finance ..... 30  
Fit-to-standard analysis ..... 227, 371, 385, 401  
Forms ..... 420  
Formula ..... 246  
Framing ..... 144  
Freestyle development ..... 166  
Free tier model ..... 58, 60, 69  
    *sign up* ..... 69  
Function ..... 179  
Function as a service (FaaS) ..... 117

## G

Gateway ..... 206  
General purpose pipeline ..... 359, 365  
Gherkin ..... 364  
Git-enabled CTS (gCTS) ..... 366  
GitHub repository ..... 111, 182, 192, 358, 379, 551  
Global account ..... 72, 76, 79  
    *entitlements* ..... 80  
    *multiple* ..... 79  
Governance ..... 238  
Government service ..... 388  
Graph ..... 268, 270  
GraphScript ..... 270  
Guided experience ..... 466

## H

Helm ..... 543  
High availability (HA) ..... 267, 346  
HTTP service ..... 198, 482  
    *access within Node.js* ..... 488  
    *build* ..... 486  
HTTP trigger ..... 179  
Human resources (HR) ... 30, 459, 461, 465, 471  
Hybrid integration platform ..... 223, 225  
Hybrid landscape ..... 368  
Hybrid supportability ..... 376  
Hyperscaler ..... 62, 65, 274, 279, 500

## I

Identity Authentication ..... 144, 325–326, 344, 351, 396  
    *configure tenant* ..... 327  
    *features* ..... 326  
Identity propagation ..... 350  
Identity provider (IdP) ..... 144, 325–326, 329, 343, 524  
    *hub* ..... 326  
Identity Provisioning ..... 326, 467  
IDoc ..... 247, 251  
iFrame ..... 144  
Implicit flow ..... 332  
    *limitations* ..... 332  
In-app extension ..... 403, 408  
    *limitations* ..... 408  
    *versus side-by-side* ..... 411  
Index server ..... 269–270  
Industry cloud ..... 31  
Information system architecture ..... 91  
Info sheet ..... 231  
Infrastructure as service (IaaS) ..... 117  
Initial risk ..... 98  
In-memory database ..... 267  
Innovation ..... 383  
Instance ..... 83, 374  
Integration ..... 38, 52, 213  
    *architecture blueprint* ..... 230

## Integration (Cont.)

<i>assess strategy</i> .....	216
<i>best practices</i> .....	228
<i>build content</i> .....	248
<i>coaching</i> .....	236
<i>developer</i> .....	235
<i>domains</i> .....	216
<i>dos and don'ts</i> .....	229
<i>guidelines</i> .....	233
<i>interfaces</i> .....	228
<i>methodology</i> .....	214
<i>monitoring</i> .....	234, 377
<i>patterns</i> .....	220, 233
<i>quality governance and assurance</i> .....	237
<i>requester</i> .....	238
<i>roles</i> .....	235
<i>scenarios</i> .....	227
<i>styles</i> .....	217–218
<i>technology mapping</i> .....	224
<i>templates</i> .....	232
<i>third party</i> .....	244
Integration Advisor .....	53, 247, 250
<i>library</i> .....	250
Integration Center of Excellence .....	235, 238–239
Integration flow (iFlow) .....	233, 435, 501, 503–504
Intelligent enterprise .....	23, 27
<i>pillars</i> .....	28
Intelligent suite .....	28
Intelligent technologies .....	35, 53
Interface .....	418
Internet of Things (IoT) .....	29, 54, 216, 399
iOS .....	169
ISO codelist .....	250
IT expert monitoring .....	234
IT operational monitoring .....	234

## J

Java .....	111
Java Database Connectivity (JDBC) .....	346
Java runtime .....	273
JavaScript .....	164
JavaScript Object Notation (JSON) .....	339
Jenkins .....	359
<i>start server</i> .....	360
Job Application app .....	488
Job management .....	377
Job server .....	293
JSON Web Token (JWT) .....	339, 341, 348
Jupyter Notebook .....	399, 455

## K

Kafka .....	254
Karma .....	364
Key criteria .....	137
Key design document (KDD) .....	402
Key performance indicator (KPI) .....	209, 405
Key user extensibility .....	403
<i>limitations</i> .....	409
Kibana dashboard .....	373
Knowledge graph .....	250–251
kubectl .....	542
Kubelet .....	125–126
Kubernetes .....	124–125, 180, 347, 542
<i>connect to on-premise</i> .....	347
Kyma .....	124, 126, 180, 183, 410, 533, 539
<i>dashboard</i> .....	540
<i>enable</i> .....	540

## L

Landscape Portal .....	531
Lean enterprise architecture ...	90, 94, 424, 446, 460, 478, 494
Lean Enterprise Architecture Toolkit .....	94
Lifecycle management .....	311
List report .....	165
Live process package .....	210
Logging .....	113, 373
Loggregator .....	123, 373
Low code/no code (LC/NC) .....	131, 172, 191

## M

Machine learning ....	35, 268, 271, 304, 396, 452
<i>algorithms</i> .....	397
<i>model</i> .....	454
<i>options</i> .....	398
Mail server .....	486
Mail task .....	206
Maintain Communication User app .....	189
Managed app router .....	160, 439
Manage KPIs and Reports app .....	405
Management console .....	293
Manual integration .....	145
Manufacturing .....	29
Mapping guideline (MAG) .....	248–249
Master data process integration .....	221
Mendix .....	131
Message-based integration .....	418
Message broker .....	254, 419
Message bus .....	116

Message filter .....	234
Message implementation guideline (MIG) .....	248
Messaging .....	123, 253
<i>resource</i> .....	119
Metadata-driven approach .....	170
Metadata Explorer app .....	285
Microservice-based architecture .....	107, 110
Microsoft Office .....	308
Microsoft Teams .....	152
Minimum viable product (MVP) .....	498
Mobile app .....	155, 157, 507–508
Mobile channel .....	134
Mobile connectivity .....	438
Mobile development kit (MDK) .....	170
Mobile frontend .....	438
Mobile integration .....	231
Mock server .....	364
Modeler app .....	284
Monetization .....	241
Monitoring .....	234, 310, 373
Monitor Visibility Scenarios app .....	207
Monitor Workflows – Workflow Definition app .....	211
Monitor Workflows – Workflow Instances app .....	206
Monolithic architecture .....	108
<i>distributed</i> .....	109
MTA descriptor .....	258
Multidimensional Expressions (MDX) .....	270
Multitarget application (MTA) .....	366
Multitarget application archive (MTAR) file .....	520
Multitenancy .....	514, 517
<i>ABAP</i> .....	530
<i>enable</i> .....	525
<i>requirements</i> .....	517
Multitenant application .....	513, 517, 519
<i>build</i> .....	520
Multithreading .....	113
MURAL template .....	99
My Inbox app .....	394, 421
My workspace .....	155

## N

Name server .....	269
Nearline storage (NLS) .....	42
Neo .....	73–74, 266, 273, 288, 336, 350, 367
Node.js .....	111, 179, 197, 488, 500
Notification event .....	252

## O

OAuth 2.0 .....	200, 330, 335, 547
<i>API</i> .....	200
<i>configure client</i> .....	549
<i>grant type flows</i> .....	330
OAuth 2.0 SAML bearer assertion .....	350
OAuth client .....	260
Object page .....	165
ObjectStore .....	280
Observed process .....	208
OData services .....	168, 180, 187, 271, 351, 353, 414, 418, 469–470, 487, 506
OData Web API .....	198
One domain model .....	395
One Page Acceptance Test (OPA5) .....	364
Online analytical processing (OLAP) ....	40, 266
Online transactional processing (OLTP) .....	40, 266
Open Connectors .....	53, 213, 243–244, 501
<i>access</i> .....	244
<i>formulas</i> .....	246
<i>functionalities</i> .....	245
Open Connectors Query Language (OCNQL) .....	245
Open Database Connectivity (ODBC) .....	346
OpenID Connect .....	335
Open integration .....	43
Open-source software .....	115
OpenUI5 .....	163
Operate phase .....	135
Operational data (O data) .....	28, 496
Operation automation .....	375
Operation concept .....	137
Order placement .....	428, 437
Order processing .....	436, 441
Order process transparency .....	426
Order status notification .....	437
Order status tracking .....	426, 429
Overview page .....	165

## P

Page type .....	165
Password grant flow .....	335
Pay-as-you-go .....	60, 69
<i>sign up</i> .....	72
Payload .....	339
Physical environment layer .....	347
Pipeline .....	285, 356, 368, 452–453
<i>ABAP environment</i> .....	359, 365–366
<i>configuration</i> .....	360

Pipeline (Cont.)  
     *extend* ..... 361  
     *general purpose* ..... 359, 365  
     *predefined* ..... 357  
     *structure* ..... 360  
 Platform as a service (PaaS) ..... 117  
 Platform technology ..... 137  
 Pod ..... 126, 544  
     *scale up* ..... 551  
 Port binding ..... 113  
 PostgreSQL ..... 274, 467  
     *administration* ..... 279  
     *architecture* ..... 275  
     *sizing* ..... 275  
 Predictive Analysis Library (PAL) ..... 268, 449  
 Predictive library ..... 271  
 Predictive model ..... 304  
 Predictive planning ..... 306  
 Prehire employee onboarding .... 461, 463, 467, 470  
 Prehire experience ..... 459  
 Price analysis ..... 446  
     *dashboard* ..... 448–449, 455  
 Principal propagation ..... 346, 352  
 Private workspace ..... 154  
 Procedure ..... 271  
 Process automation ..... 390  
 Process flexibility ..... 209  
     *cockpit* ..... 210  
 Process innovation ..... 389  
 Process integration ..... 217–219, 261, 390  
 Process model ..... 389  
 Process template ..... 208, 389  
 Process visibility ... 207, 394, 429, 432, 441, 466  
 Pro-code ..... 131  
 Procurement ..... 29  
 Procurement spend ..... 443  
 Product browsing ..... 428, 437  
 Product experience ..... 31  
 Product information ..... 435  
 Production ..... 75  
 Profile page ..... 154  
 Programming model ..... 133, 185, 410  
     *comparative review* ..... 190  
 Project “Piper” ..... 114, 357, 359, 362, 368, 379, 543  
     *testing* ..... 365  
 Project implementation ..... 363  
 Project lifecycle ..... 93  
 Proof Key for Code Exchange (PKCE) ..... 332  
 Proof of concept (PoC) ..... 236  
 Provider manager ..... 146, 150  
 Public workspace ..... 154

## Q

Quality assurance ..... 238  
     *requirements* ..... 239  
 Quality gate management (QGM) ..... 368  
 Quality specialist ..... 238  
 Qualtrics ..... 30, 463, 465, 493, 495, 501  
     *action platform* ..... 503  
     *connect to SAP BTP* ..... 502  
     *prerequisites* ..... 501  
     *surveys* ..... 208  
 Query language (QL) ..... 272  
 Query optimizer ..... 272  
 Questionnaire ..... 239  
 Queue subscription ..... 253  
 QUnit ..... 364

## R

Reactive architecture ..... 118  
 Recipient list ..... 234  
 Refresh token flow ..... 334  
 Regression ..... 304  
 Reinforced learning ..... 397  
 Relational database management system (RBDMS) ..... 42, 266  
 Remote function call (RFC) ..... 201  
 ReplicaSet ..... 126  
 Reporting ..... 297, 301, 416  
 Reports, interface, conversion, enhancements, forms, and workflow (RICEFW) ..... 415  
 Residual risk ..... 98  
 Resource unit (RU) ..... 120  
 Response variable ..... 196  
 Reverse invoke proxy ..... 344  
 Rideshare app ..... 497  
 Risk analysis ..... 98  
 Risk index ..... 477–479, 482, 487  
 Roadmap ..... 106  
 Robotic desktop automation ..... 54, 391  
 Robotic process automation (RPA) ..... 54, 134, 390  
 Role ..... 77, 342  
     *assign* ..... 77  
     *collections* ..... 77, 342, 344, 358, 367, 524  
     *integration* ..... 235  
     *templates* ..... 342  
 Role-based access ..... 341  
 Rollback ..... 126  
 Rolling update ..... 126  
 Root cause analysis ..... 377  
 Route ..... 527

Router ..... 122  
 Rule ..... 195  
     *services* ..... 194  
 Ruleset ..... 195  
 Run and scale phase ..... 94  
 Runtime ..... 121, 176  
     *comparative review* ..... 183

## S

SAML ..... 351  
 SAML bearer assertion ..... 352  
 SAML bearer token ..... 144  
 SAP Activate ..... 227, 372, 401  
 SAP Adaptive Server Enterprise (SAP ASE) ..... 42, 266, 273  
 SAP AI Business Services ..... 54, 398  
 SAP Analytics Cloud ..... 34, 50, 297, 299, 400, 417, 447, 450, 452, 455–456  
     *business intelligence* ..... 49  
     *capabilities* ..... 300  
     *dashboards* ..... 448  
     *data connectivity* ..... 308  
     *mobile* ..... 308  
     *platform* ..... 308  
     *portfolio* ..... 300  
     *smart features* ..... 303  
     *user interfaces* ..... 307  
     *versus SAP BPC* ..... 318  
     *versus SAP BusinessObjects BI* ..... 319  
     *versus SAP Data Warehouse Cloud* ..... 320  
     *versus SAP IBP* ..... 319  
 SAP Analytics Cloud for planning ..... 50  
 SAP API Business Hub .... 52, 129, 206, 240, 242, 244, 254, 257, 435  
     *access connectors* ..... 244  
 SAP AppGyver ..... 131, 172, 175, 432, 438, 500, 507  
 SAP Application Extension  
     *Methodology* ..... 136  
 SAP Application Interface  
     *Framework* ..... 226, 234  
 SAP Ariba ..... 29, 395, 443–445, 447, 452–453  
 SAP Ariba Cloud Integration  
     *Gateway* ..... 444, 451  
 SAP Authorization and Trust Management  
     *Service* ..... 488  
 SAP Automation Pilot ..... 374–375, 379  
 SAP BTP, ABAP environment ..... 135, 182–183, 188, 356, 359, 417, 482, 484–486, 488, 513, 519, 534  
     *call business rules* ..... 198  
     *multitenancy* ..... 530

SAP BTP, Cloud Foundry environment ..... 135, 145, 288, 336, 341, 344, 352, 367, 534  
 SAP BTP, Cloud Foundry runtime ..... 121, 123, 136, 176, 183, 456, 500, 537  
     *unsupported features* ..... 123  
 SAP BTP, Kyma runtime ..... 124, 126, 134–136, 180, 183, 533–534, 537–538  
     *activate* ..... 539  
     *application deployment* ..... 542  
     *components* ..... 181  
     *configure OAuth client* ..... 549  
     *create API rule* ..... 547  
     *deploy services* ..... 544  
     *initial setup* ..... 539  
     *scalability* ..... 550  
 SAP BTP, Neo environment ..... 273, 288, 336, 350, 352, 367  
 SAP BTP, serverless runtime ..... 179, 183, 534  
 SAP BTP cockpit ..... 69, 77, 80, 258, 357, 539  
     *authorization* ..... 343  
 SAP Business Application Studio ..... 133, 155, 168, 182, 356, 379, 469–470, 489, 523  
 SAP Business Network ..... 30, 387, 447, 452  
 SAP Business Network for Suppliers ... 444, 447  
 SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI) ..... 49, 319  
 SAP Business Planning and Consolidation (SAP BPC) ..... 50, 297, 299  
     *versus SAP Analytics Cloud* ..... 318  
 SAP Business Process Management (SAP BPM) ..... 262  
 SAP Business Rules Management (SAP BRM) ..... 262  
 SAP Business Technology Platform (SAP BTP) ..... 23, 31–32, 57, 446  
     *account setup* ..... 75  
     *account structure* ..... 72  
     *administration* ..... 76  
     *analytics* ..... 295  
     *architecture* ..... 89  
     *authentication flows* ..... 336  
     *business logic* ..... 175  
     *capabilities* ..... 38  
     *connect to on-premise* ..... 345  
     *create account* ..... 67  
     *database and data management* ..... 265  
     *development approach* ..... 127  
     *DevOps* ..... 355  
     *environments* ..... 73  
     *integration* ..... 213  
     *launchpad* ..... 135  
     *models* ..... 57, 60

SAP Business Technology Platform (Cont.)	
<i>pillars</i> .....	33, 36, 39
<i>runtimes</i> .....	121
<i>security</i> .....	323
<i>services</i> .....	61
<i>trial</i> .....	58, 60, 67, 79
<i>types of innovation</i> .....	383
<i>user interface</i> .....	141
SAP Business Warehouse (SAP BW) .....	318
SAP BW/4HANA .....	50, 298, 318
<i>versus SAP Data Warehouse Cloud</i> .....	321
SAP Cloud ALM .....	370, 376
<i>deployment orchestration</i> .....	372
<i>implementation pillar</i> .....	370
<i>versus SAP Solution Manager</i> .....	372
SAP Cloud Application Programming Model .....	111–112, 133, 175, 185, 190, 267, 274, 410, 414, 428, 432, 437, 450, 452, 469, 500, 502, 513, 537–538
<i>build a multitenant application</i> .....	520
<i>build microservice</i> .....	437, 506
<i>build multitenant applications</i> .....	516
<i>CI/CD</i> .....	362
<i>connectivity isolation</i> .....	518
<i>data isolation</i> .....	517
<i>enable multitenancy</i> .....	525
<i>principles</i> .....	186
<i>tenant awareness</i> .....	516
SAP Cloud Integration for data services .....	263, 289
<i>agent</i> .....	290
<i>data flow</i> .....	292
<i>data store</i> .....	291
SAP Cloud Platform .....	39
SAP Cloud Portal service .....	158
SAP Cloud SDK .....	414
SAP Cloud Transport Management .....	358, 366, 368, 372, 379
<i>CI/CD</i> .....	362
SAP Concur .....	394, 477, 479, 481
SAP Connectivity service .....	257, 347, 451
SAP Continuous Integration and Delivery .....	368
SAP Continuous Integration and Delivery .....	357–358, 362, 379, 539, 543, 551
<i>initiate build process</i> .....	358
<i>role</i> .....	358
SAP Conversational AI .....	54, 134, 155, 214, 432, 439
SAP Crystal Reports .....	298
SAP Crystal solutions .....	49
SAP Customer Experience .....	213, 493, 495, 497, 501, 503–504
<i>prerequisites</i> .....	501
SAP Data Hub .....	44
SAP Data Intelligence .....	44, 48, 263, 280–281, 297, 398, 450, 452–454
<i>features</i> .....	282
SAP Data Quality Management, microservices	
for location data .....	46
SAP Data Services .....	45, 47–48, 293
SAP Data Warehouse .....	45
SAP Data Warehouse Cloud .....	34, 49, 297–298, 312, 314, 317
<i>capabilities</i> .....	316
<i>data connectivity</i> .....	313
<i>integration</i> .....	313
<i>versus SAP Analytics Cloud</i> .....	320
<i>versus SAP BW/4HANA</i> .....	321
SAP Destination service .....	257, 336, 345, 484, 502, 549
SAP Digital Boardroom .....	300, 307
SAP Discovery Center .....	62
SAP Edge Services .....	55
SAP Enterprise Information Management (SAP EIM) .....	48
SAP Enterprise Portal .....	159
SAP ERP .....	401
SAP Event Mesh .....	119, 130, 179, 251, 429, 431, 434, 437, 538
<i>communication patterns</i> .....	253
SAP Extension Suite .....	35, 51, 128, 191, 410
<i>capabilities</i> .....	51
SAP Fiori .....	135, 159, 161, 439, 469
<i>apps</i> .....	165, 508
<i>CI/CD</i> .....	362
<i>design principles</i> .....	161
<i>tools</i> .....	164, 168
SAP Fiori Cloud .....	159
SAP Fiori elements .....	162, 164, 166, 168, 417, 506
<i>page types</i> .....	165
SAP Fiori launchpad .....	135, 159, 414, 546
SAP Focused Run .....	377
SAP Forms service by Adobe .....	420
SAP Gateway .....	187, 351–352
SAP Graph .....	255, 395
<i>API</i> .....	256
SAP HANA .....	34, 40, 50, 257, 266, 285, 288, 297–298
<i>architecture</i> .....	269
<i>features</i> .....	267
<i>technical capabilities</i> .....	270
SAP HANA Cloud .....	34, 41, 50, 266, 288, 297–298, 317, 428, 433, 449, 453–454, 456, 466, 484, 500, 538
<i>architecture</i> .....	269

SAP HANA Cloud (Cont.)	
<i>data lake</i> .....	268
<i>features</i> .....	267
<i>technical capabilities</i> .....	270
SAP HANA Deployment Infrastructure (HDI) container .....	454, 517, 523
SAP HANA extended application services, advanced model (SAP HANA XSA) .....	267, 269
<i>CI/CD</i> .....	362
SAP HANA extended application services (SAP HANA XS) .....	267, 269, 272
SAP HANA smart data integration (SDI) .....	286
SAP HANA smart data quality (SDQ) .....	289
SAP HANA XS JavaScript (XSJS) .....	272
SAP Help Portal .....	228
SAP ID service .....	327
SAP Information Lifecycle Management (SAP ILM) .....	47
SAP Information Steward .....	44
<i>scenarios</i> .....	45
SAP Integrated Business Planning (SAP IBP) .....	29, 289, 319
SAP Integration Solution Advisory Methodology .....	214, 228
<i>assess integration strategy</i> .....	216
<i>blueprints</i> .....	230
<i>decision tables</i> .....	225
<i>define best practices</i> .....	228
<i>design hybrid integration platform</i> .....	223
<i>enable practice of empowerment</i> .....	234
<i>info sheets</i> .....	231
<i>integration styles</i> .....	217
<i>templates</i> .....	232
<i>use case patterns</i> .....	220
SAP Integration Suite .....	35, 52, 213, 224, 226, 262, 390, 501
SAP Intelligent Business Process Management (SAP Intelligent BPM) .....	389
SAP Intelligent Robotic Process Automation (SAP Intelligent RPA) .....	35, 54, 134, 391, 449, 452, 454, 456
SAP Internet of Things (SAP IoT) .....	54, 400
SAP IQ .....	41
SAP Jam .....	159
SAP Job Scheduling service .....	482, 484, 488
SAP Landscape Transformation Replication Server .....	47, 293
SAP Launchpad service .....	142, 545
<i>access and authentication</i> .....	144
<i>integration</i> .....	144
<i>tools and services</i> .....	146
<i>versus SAP Work Zone</i> .....	156
SAP Manufacturing Integration and Intelligence (SAP MII) .....	399
SAP Master Data Governance (SAP MDG) .....	43
SAP Mobile Services .....	155, 170, 232, 432, 438, 500, 509
SAP Mobile Start .....	157
SAP OData Provisioning .....	180
SAP partner .....	37, 72
SAP PartnerEdge .....	72
SAP PowerDesigner .....	46
SAP Process Integration .....	261, 418
SAP Process Orchestration .....	225–226, 261, 418
SAP S/4HANA .....	30, 142, 163, 200, 393, 401, 403, 412, 416, 427
<i>connect to SAP BTP</i> .....	434
SAP S/4HANA Cloud .....	128, 213, 427
<i>connect to SAP BTP</i> .....	433
<i>extensibility</i> .....	431
SAP Sales Cloud .....	256
SAP Screen Personas .....	172
SAP Service and Asset Manager .....	171
SAP Signavio .....	389
SAP Software-as-a-Service Provisioning service .....	516
SAP Solution Manager .....	362, 368–369, 376–377
<i>versus SAP Cloud ALM</i> .....	372
SAP SQL Anywhere .....	42
SAP SuccessFactors .....	156, 378, 459–462, 467, 479, 481
<i>events</i> .....	254
<i>OData service</i> .....	487
SAP SuccessFactors Work Zone .....	466–467, 474
SAP Supply Chain Control Tower .....	29
SAPUI5 .....	162–163, 432, 466, 469, 484, 489
<i>CI/CD</i> .....	362
<i>testing</i> .....	363
SAP Web IDE .....	242
SAP Workflow Management .....	155, 175, 191, 205, 394, 421, 429, 431, 436, 464, 466–467, 484, 489, 538
<i>audit trail</i> .....	207
<i>business rules</i> .....	193
<i>process flexibility</i> .....	209
<i>process visibility</i> .....	207
<i>wizard</i> .....	192
<i>workflow modeling</i> .....	205
SAP Work Zone .....	67, 103, 135, 151–152, 388, 429, 433, 439, 463, 467, 473, 545
<i>capabilities</i> .....	152
<i>extensibility</i> .....	155
<i>mobile apps</i> .....	155



SAP Work Zone (Cont.)	
<i>project templates</i>	156
<i>versus SAP Launchpad service</i>	156
SAP Work Zone for HR	156
Scalability	109–110, 116, 119, 267, 379, 550
<i>horizontal</i>	537
Scatter-gather	234
Scheduling	311
Schema definition language (SDL)	272
Schema separation	515
Scope	342
Script task	206
Search to insight	303
Secure Token Service (STS)	352
Security	267, 323
<i>descriptor</i>	341, 343
<i>flows</i>	329
<i>supported configurations</i>	337
Security Assertion Markup Language (SAML)	144, 325, 329
<i>token</i>	329
Self-service analytics	298
Semisupervised learning	397
Serverless architecture	117, 127
Serverless runtime	179
Service binding	527
Service catalog	127, 182
Service consumption	189
Service instance	83
Service-layer security	346
Service manager	523
Service marketplace	83, 358
Service plan	81
Services discovery	61, 65
<i>add</i>	66
Service task	196, 206
Service ticket	503–504
Service-to-service communication	118
Session management	270
Side-by-side extensibility	129, 378, 402, 410
<i>versus in-app</i>	411
Signature	340
Simple Mail Transfer Protocol (SMTP)	483
Single-service subscription	59
Single sign-on (SSO)	144, 230, 326
Site	146
<i>alias</i>	146
<i>directory</i>	146
<i>settings</i>	147
Smart data integration	286
Smart discovery	304
Smart grouping	455
Smart insights	304
Smart predict	304, 306
Smart sensing	400
smsi tool	523
Software as a service (SaaS)	117, 385
Software development kit (SDK)	311
Software distribution	104
Solution concept diagram	102, 427, 447, 463, 481, 497, 536
Solution context diagram	98, 445, 461, 479
Solution realization diagram	102, 428, 464, 482, 498, 537
Space	315
Splitter	234
Spreadsheet upload	419
SQL	270
SQLite	466
SQLScript	271
Stakeholder management	96
Stakeholder matrix	96
Standalone app router	160
StatefulSet	126
Stateless services	112
Statement of architecture work	90, 97
Story	300, 307
Strategy map	95, 424, 444, 460, 478, 494
Structure	194
Subaccount	72, 75, 80, 520, 539
<i>assign entitlements</i>	81
<i>assign service</i>	83
<i>connect backend systems</i>	128
<i>create</i>	80
<i>environment</i>	73
<i>trust setup</i>	327
<i>view entitlements</i>	83
Subscription plan	83
Supervised learning	397
Supplier analysis dashboard	450, 455
Supplier rating	446, 448
Supply chain	29, 444
Survey project	501
Survey results	503
Sustainability management	32
<b>T</b>	
Table	194
<i>access</i>	530
Task	206
Technology architecture	92

Technology driver	92
Technology pillars	33, 39
Tenant awareness	514, 516, 526, 530
Tenant identification	517
Tenant isolation	530
Tenant offboarding	517
Tenant onboarding	517
Test automation	363
Testing	238, 251, 363, 506
Text search	268
The Open Group Architecture Framework (TOGAF)	90
Thing integration	218
<i>patterns</i>	223
Timer trigger	179
Time series	304
Token-based security	116
Token service URL	550
Trace analysis	377
Transaction	
<i>/IWFND/MAINT_SERVICE</i>	352–353, 417
<i>CATSXT</i>	388
<i>CERTRULE</i>	353
<i>OA2C_CONFIG</i>	201
<i>PFCG</i>	352
<i>SAML2</i>	352
<i>SE80</i>	200
<i>SEGW</i>	188, 417
<i>SM59</i>	201, 257, 336, 531
<i>SOAUTH2</i>	352
<i>STRUST</i>	201, 353
<i>SU01</i>	325
Transport node	369
Trial account	57, 79
<i>create</i>	67
<i>types</i>	58
Trigger	179
Tunneled access	144
Twelve-factor principle	111, 116
Two-factor authentication	326
Type safe	414
Type system	250
<b>U</b>	
UI5	163
UI integration card	155, 466
UIVeri5	365
UN/CEFACT	251
UN/EDIFACT	250
Unattended mode	391
Unmanaged app router	160
Unsupervised learning	397
Use case	
<i>architecture</i>	462, 480, 496, 514, 536
<i>blueprint</i>	99
<i>build cloud-native applications using Kyma runtime</i>	533
<i>building multitenant applications</i>	513
<i>business gaps</i>	400
<i>extend and integrate SAP S/4HANA</i>	423
<i>extending functionality</i>	386
<i>extending reach</i>	387
<i>extend SAP Ariba</i>	443
<i>extend SAP SuccessFactors</i>	459
<i>identify</i>	383
<i>implementation</i>	433, 453, 467, 485, 501, 520, 539
<i>integrate SAP Customer Experience with Qualtrics</i>	493
<i>Integrating risk index with SAP Concur</i>	477
<i>patterns</i>	220
<i>process innovation</i>	389
User account and authentication (UAA)	339, 528
User base	388
User experience (UX)	141, 161, 418
User integration	217–218
<i>patterns</i>	223
User interface (UI)	141
<i>adaptation</i>	404
<i>development tools</i>	161
User management	325–326
User monitoring	377
User role	77
User task	206
<b>V</b>	
VCAP	112
Virtualization	268, 313
Visibility scenario	208, 211, 441
Visualization	301
Visual Studio Code	133, 168
<b>W</b>	
Waterfall-based development	363
Web channel	134
Web component	164
Webhook	379
Web service endpoint	503



Workflow ..... 196, 205, 394, 420, 432, 436, 464, 466, 468

*add scope* ..... 206

*modeling* ..... 205

Work package ..... 106

Workspace ..... 153

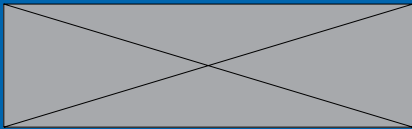
*templates* ..... 466

*type* ..... 154

X

X.509 certificate ..... 144, 352

XSODATA ..... 271



Smitha Banda, Shibaji Chandra, Chun Aun Gooi

## SAP Business Technology Platform: An Introduction

570 Pages, 2022, \$79.95  
ISBN 978-1-4932-2201-8

 [www.sap-press.com/5440](https://www.sap-press.com/5440)



**Smitha Banda** is a senior business development manager at SAP and leads the SAP BTP practice for North America. She started her journey at SAP in 2005 as a consultant providing advisory, architecture, and development services to customers across the globe. She has always worked closely with customers and continues to do so in her current role.



**Shibaji Chandra** is a seasoned consulting architect at SAP who has in-depth knowledge and expertise in SAP BTP. Shibaji has been working in the SAP space for more than 18 years and has extensive work experience with SAP's cloud platforms and related services since 2013. Throughout the years, he has worked very closely with customers, provided solution architectures, and defined best practices, strategies, and methodologies.



**Chun Aun Gooi** is an SAP BTP architect and a developer at heart. He has more than 10 years of SAP experience in architecting, building, and enhancing enterprise software. He has been involved in a number of customer projects, built several proof of concepts for customers and reference architectures, and provided guidance and strategy to customers on various SAP BTP topics.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.