Browse the Book

This chapter provides an overview of digital transformation and introduces how Agile innovation triggered the evolution and the need for SAP Cloud Platform. It also describes the fundamentals of cloud-native application development, along with introduction of cloud platforms, microservices, deployment options, and more.

“SAP Cloud Platform: Digital Transformation and Innovation”

Table of Contents

Index

The Authors

Gairik Acharya, Govind Bajaj, Avijit Dhar, Anup Ghosh, Asidhara Lahiri

SAP Cloud Platform: Cloud-Native Development

489 Pages, 2019, $79.95


www.sap-press.com/4713
Chapter 1
SAP Cloud Platform: Digital Transformation and Innovation

In this chapter, we’ll explore how digitization is changing the business world and increasing the demand for Agile innovation. Then, we’ll introduce SAP Cloud Platform as the environment capable of meeting the needs of modern-day enterprises.

Globally, enterprises in practically all industries are facing unprecedented situations. Innovation, whether through new technologies or new business models or through using old technologies in new ways, is emerging at a faster rate than ever before. Current industry leaders are being displaced by competition that leverages these innovations to reshape entire industries, wiping away anything that comes in its path. The only way enterprises can handle assaults from the competition—emerging from not only in the same industry but also in adjacent industries, from digital upstarts, and from rapidly diversifying giants—is through continuous reinvention. Enterprises should innovate not only when major tech or business inventions arise but also even during apparently calm status quo periods when things seem absolutely fine.

Enterprises need IT platforms to support this continuous transformation and reinvention. In this book, we'll evaluate how SAP Cloud Platform can help in this journey. But, first, let's look more at the digital transformation and how cloud platforms can help.

1.1 Digital Enterprise: What and Why of Digital Transformation

An enterprise leveraging the power of the latest digital technologies to further its business interest can be called a “digital enterprise.” While digitization of business processes started decades ago through the adoption of information technologies focused mainly on recording and automating otherwise manual activities, an inflection point has now been reached where new business models are emerging by leveraging digital technologies. Different enterprises are at different stages of this
continuum, spanning from basic digitization to advanced capabilities. This transformation is colloquially known as “digital transformation.” The digital transformation of an enterprise involves the following steps:

1. Defining business objectives
2. Assessing current state of digitization
3. Developing a roadmap from the current state to the target state

While these steps sound like the traditional plans where a target five years ahead is set and achieved, the volatility in the current business environment often makes five-year targets obsolete much earlier. The main challenge for enterprises is to strike the right balance in a future outlook grounded in the present.

In these difficult times, enterprises can go back to basics of good business. Outstanding enterprises focus on three things:

- Knowing clients as individuals
- Partnering with the best
- Empowering employees

In the past, C-suite executives could see the competition coming and counter the competition with better, cheaper, and more competitive products. Now, the market is so volatile that the competition is often not even noticed until they’ve already been uprooted.

For the first time, the entire C-suite recognizes technology as the most impactful game changer. Realizing that the competition will leverage technology to disrupt the leading enterprises, executives now proactively plan to leverage technology themselves to know their clients better, to partner with the best, and to empower employees. They’ll start redrawing the rules of engagement with their customers, partners, and employees. Instead of depending entirely on homegrown innovations, external partners can bring about innovation. They also recognize the need for more decentralized decision-making closer to the customer and action to combat the nimble and agile competitor and leverage the power of partner network.

Thus, enterprises should focus on developing capabilities to sustain longer-term activities than functionalities that may need to change faster. Thus, enterprises should focus on having a digital core in place to support their core, stable business processes, with capabilities to extend and collaborate as required. The flexibility to handle the unknown and the ability to handle the speed of change are two capabilities that all enterprises need to embed into their solutions.

Digital transformation should add these capabilities to an enterprise. Of course, basic hygiene factors, like lowering the total cost of ownership of IT to free up resources to focus on adding differentiating capabilities, will continue to underpin digital transformation. As IT transforms its role from being a way to lower bottom lines to an enabler of increasing top lines, now IT must play both roles. In other words, changing fast (developing fast) while running cheap is the new focus.

Digital innovation enables enterprises to reach customers and consumers in a completely new way, resulting in better customer satisfaction and thus a better business. Their improved understanding of individual consumers’ needs allows an enterprise to offer better products and services. Newer digital technologies allow better collaboration with partners and thus the opportunity to develop newer offerings leveraging the power of many and helping employees feel empowered.

In short, digital transformation is about:

- Increased automation of stable core processes to reduce time-consuming, error-prone, and costly manual activities to help lower costs, increase efficiency, improve quality, and free up premium resources for better work
- Introducing quick innovation to enable newer business processes and business models to develop new markets and increase revenues, which enables enterprises to develop new markets and increase revenues

1.1 Digital Enterprise: What and Why of Digital Transformation

Digital transformation is a journey of doing business in a different way, often in newer domains in newer industries adopting latest digital technologies. This transformation allows enterprises to reevaluate their missions, review their objectives, redesign their business models, reform their processes, revamp the way they interact with their customers/partners, and reignite the passion of their employees. These changes are often not incremental, but rather revolutionary. Thus, we’ve seen enterprises venturing into completely new domains or completely a new set of players dominating certain industries. Thus, the impact of the transformation is far reaching both inside and outside an enterprise.

Thus, an agriculture equipment manufacturer becomes a large IT company; a video rental company becomes one of the largest producers of movies; an enterprise with no rooms becomes the largest hotelier company; a company without cars becomes the largest transportation provider. While different eras of modern human history have seen several transformation agents, from the discovery of fire to the discovery of the steam engine and beyond, the current confluence of digital technologies has
the potential to create similar impact. While still early, a world of untapped potential is on the horizon.

From predicting what the next potential customer will buy, which will improve business revenue, to predicting cancer, which can save lives, the advancement of digital technologies is happening in different areas. Together, these technologies can impact so many spheres of life that no enterprise can afford to ignore them. While several of these technologies—namely, cloud computing, analytics, mobility, social, blockchain, the Internet of Things (IoT), machine learning, artificial intelligence, virtual reality, etc.—emerged at different times and individually have the potential for significant impact, their combined power is what makes the most significant impact. The time is right in terms of the maturity of these technologies and the readiness of the ecosystem, as we’ll discuss in detail throughout this chapter.

What started as a way to record data and automate repeatable manual activities resulted in the generation of data, both structured and unstructured, which subsequently opened enormous possibilities, starting the evolution from basic digitization to advanced digital transformation. The ability to make use of this data to understand customers, partners, employees, events, trends, etc. has opened up new horizons for enterprises. More accurate and impactful decisions can be made after the fact as well as at the moment a transaction is happening, and enterprises can predict what will happen so that the right decisions can be made. Cheap and easy storage technology, powerful on-demand computing capacity, and advanced mathematical techniques allow enterprises to record and store a lot more data than was possible in the past, analyze and contextualize this data quickly, and support decision-making. These capabilities, along with seamless connection with people and devices from other enterprises, customers, partners, etc., form the fundamental technical underpinning of digital transformation. Systems and solutions are becoming more intelligent, smarter, and more cognitive—that is, they are becoming more context aware self-learning. Digital transformation leverages these powers to boost business. While technology is an enabler, it is not the only lever of this transformation. Technology must be supported by changes in business processes and employees. Enterprises that successfully navigate this journey will orchestrate these dimensions in a way to maximize the power of transformation.

1.1.2 Path to Digital Transformation
As discussed earlier, digital transformation is not a one-and-done activity. This journey has several potential paths, depending on enterprise priorities as enterprises move from their current state. Different enterprises will follow different paths with the same objective of transforming business by leveraging digital technologies.

Each company would have its own vision, its own value drivers, its own unique needs, and of course its own pain points. Use cases should be identified to initiate the transformation. Unlike in the past when enterprises developed five-year plans, which may be obsolete before they are implemented, a use case-based approach provides the ability and flexibility to adopt and adapt to changing business needs. Quick wins can be prioritized by implementing use cases that provide immediate benefits.

Often, clients must first ensure that their core backbone systems are upgraded to form a digital core to serve as the foundation for incorporating advanced capabilities like blockchain, IoT, advanced analytics, etc. subsequently, at the appropriate time. This incremental approach prevents the need for a big bang implementation, which requires large investments in capital and human resources and can involve major disruption.

1.2 The Role of the Cloud in Digital Transformation
As discussed earlier, to support digital transformation, speed, flexibility, high quality, and low cost are absolute musts. The ability to introduce these qualities is often the key success factor in a digital transformation project.

The speed to release new features/functions/use cases, the speed to market new capabilities, and the speed of adoption are essential for success. Traditional models of long-term planning and execution are no longer relevant in this fast-moving world. If an enterprise needs to change its business model or reach out to customers, it needs to do so quickly. To stop fraud, a company must analyze data quickly, before a transaction happens. Thus, its IT infrastructure must be scalable and on demand. Instead of developing IT solutions themselves, companies can leverage offerings from software as a service (SaaS) providers, instead of waiting for systems/software to be procured, installed, and configured/extended to implement functionalities. Enterprises can start configuring/extension immediately by leveraging platform as a service (PaaS) or SaaS from appropriate providers. By not owning the software, companies have the flexibility to change if required. By also not owning the hardware and instead using SaaS or PaaS services on demand from cloud providers, the total cost of ownership is lowered and flexibility increased. This cost savings and increased flexibility provide the necessary support for maximizing the value of a digital transformation.
The cloud can also help improve quality by providing standardization. In short, the cloud provides the following benefits:

- **Speed**
  Faster provisioning of hardware software, faster development lifecycle, faster release

- **Flexibility**
  Ease of change

- **Quality**
  Better predictable quality through standardization

### 1.2.1 Cloud Variants

As discussed earlier, the cloud comes in different flavors based on the following:

- **Infrastructure as a service**
  In the infrastructure as a service (IaaS) variant, hardware, networks, etc. are consumed from a wide range of possible hosting arrangements, from a private dedicated server in your own data center to a completely shared and public cloud.

- **Platform as a service**
  In the platform as a service (PaaS) variant, software tools, along with the necessary hardware, are combined and provided as a service to allow you to develop and run applications quickly without the need to build your own infrastructure, software stack, etc.

- **Software as a service**
  With software as a service (SaaS), clients can consume applications without worrying about infrastructure.

Increasingly, enterprises consume variations of each of cloud variant and have multiple clouds in their portfolio. Thus, most enterprises would have a hybrid multicloud infrastructure—in other words, a mixture of on-premise systems and private and public clouds for IaaS, while consuming PaaS and SaaS offerings from multiple appropriate service providers.

### 1.2.2 How Cloud Supports Digital Transformation

As discussed earlier, the core to digital transformation is introducing flexibility, agility, and speed to the enterprise. The cloud, in all its variants, can help in that mission. IaaS helps set up the right infrastructure quickly at competitive prices without having to wait days for expensive hardware to arrive and without days of effort for setting up the system. If the hardware is needed only when workloads peak or when test cycles are conducted, then the enterprise doesn’t need to pay for the unutilized time, doesn’t need to hoard data center floorspace, and doesn’t need to worry about getting rid of old hardware when new hardware is required. If special compliance needs do not require using a private dedicated infrastructure in some specific data center, an enterprise can gain flexibility, speed, and savings by leveraging shared public clouds from reliable cloud service providers. A server can be implemented only when required, only for the time required, and then can be discarded when not needed. By extension, managed services on an infrastructure often bundle up infrastructure-related services with software management services e.g., managing both the operating system and the database and, in the case of SAP, a managed SAP Basis layer as well. In this case, you can use SAP HANA Enterprise Cloud, which relieves some of the pain in managing these services or through contracted third-party vendors. As a result, projects will enjoy a fast start.

Enterprises now have the option of leveraging PaaS to develop an application or an extension to existing applications or to add certain IT or business capabilities. Expensive software, and the corresponding hardware, won’t need to be procured or configured before the development team can start development. Built-in development tools, DevOps toolchains, and many other software development lifecycle (SDLIC) tools enhance the speed and quality of development. SAP anticipated this need for enterprises running SAP and others and thus came up with its PaaS offering: SAP Cloud Platform, which is one of the most important products in SAP’s portfolio. While not limited to existing SAP customers, SAP Cloud Platform provides the right development and integration platform to facilitate the development of extensions to SAP solutions and to introduce important digital capabilities like the Internet of Things (IoT), blockchain, mobility, etc. in an easy, fast, and convenient way.

Enterprises can now consume fully functional applications for human resources, supply chain management, and more over the cloud in days instead of the many months of preparation, procurements, installation, massive configurations/development, and so on, as in the past.

Thus, the cloud is a key enabler of an enterprise’s digital transformation journey by providing the necessary agility, flexibility, and speed.

In most cases, enterprises will end up with a mixture of all three variants, resulting in a hybrid multicloud situation with a gradual transition from private dedicated on-premise systems to public multicloud systems.
1.2.3 Key Considerations

In this journey, enterprises will need to make some decisions to ensure that the transformation is right for their organization, at the right pace and in the right direction. Thus, each journey is different for each organization.

Which applications should stay within a secured dedicated data center and which can be outsourced to a service provider must be carefully evaluated as part of strategic decision-making.

The question, “Can I trust my data with somebody else?” is often paramount and a key deciding factor. Enterprises can evaluate the security policies of cloud service providers to ensure their data is safe. Technical and organizational measures in place to comply with regulatory requirements like the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), etc., can be evaluated. The different data center locations, points of presence (POPs), and points of deliveries (PODs), can be evaluated to ensure adequate geographical coverage. Capabilities like bring your own image, bring your own IP can often play a role in the decision. Policies around data aggregation (whether data will be used for analytics along with data from other enterprises even at an aggregate level, policies around tenant prioritization, level of self-service provided, level of automation provided) play nontrivial roles in this decision-making. High availability and disaster recovery, along with different service level agreements, are key considerations.

For PaaS, enterprises can evaluate the richness of available services and functions in the platform, from the availability of highly scalable, secure container management systems to integration services for machine learning, IoT, blockchains, etc., so that enterprises can adopt these capabilities easily when needed. Lifecycle management services should be built in to toolchains for DevOps and other relevant practices that can be leveraged in a standardized way across the enterprise. The level of automation (versus manual effort) is also a key differentiator when evaluating cloud service providers.

While an enterprise might want service providers to take care of everything required to provide those services, most enterprises would want visibility into what is happening and have a say in the control of these activities. Thus, integrated dashboards and other monitoring tools are important for feedback and the ability of the service provider to act quickly on feedback. Similarly, the interoperability between the PaaS and other platforms, as well as extensibility features so enterprises can adopt/extend/develop new features, plays a key role as well.

Application software as a service (SaaS) providers can be evaluated on their security capabilities, their ability to truly segregate and isolate multiple clients using the same system, their upgrade strategies, their extensibility capabilities, and their integration capabilities.

Since most enterprises end up with mix of all these functions in their portfolio, cloud service providers must adhere to a unifying architecture that covers both on-premise and cloud-based capabilities. Enterprises must also account for different providers under the same umbrella, which can lead to a unifying architecture that the enterprise would follow.

Enterprises should also assess the long-term potential of their cloud providers to ensure that these providers would continue to provide services over a long period of time. It’s often a judgment call between adopting breakthrough innovations a new entry may provide versus the stability that established providers offer.

An enterprise must undertake a big shift in mindset in terms of adhering to standards when adopting the cloud in their digital transformation journey. The true value of the cloud may be realized when an enterprise can consume standard offerings as much as possible, which reduces risk, costs, and time to procure. More custom requirements start to erode the value. Thus, the optimal balance between custom versus standard deployments must consider precisely when/to where to develop custom capabilities, how to manage their lifecycles, etc. which is required.

In the context of SAP-centric landscapes, a good practice is to keep the core SAP systems as standard as possible. Any required customization should be built on a development platform (e.g., SAP Cloud Platform) decoupled from the core, which will reduce impact of upgrading in the future. Serious attempts should be made to convert custom functions into APIs (application programming interfaces) and microservices for more standardized reuse and ease of maintenance along with ability to scale up. Containerization of custom solutions helps make deployment and scaling up easier. Custom solutions should be defined, designed, and developed in completely new ways to maximize the value from the cloud.

Given its unique nature, in the past, an SAP landscape had to be isolated from the generic trends elsewhere in the IT domain. However, in keeping with the changing requirements of the customers, SAP has embraced openness, interoperability, and standardization. In fact, SAP participates in and sponsors several initiatives with other players. The result is an enormous opportunity for SAP-centric enterprises to lead this digital transformation journey with a solid core of proven best practices in the form of SAP ERP or SAP S/4HANA, along with the flexibility and openness to integrate with others, adopt newer technologies. Thus, SAP Cloud Platform today offers services like IoT, machine learning, blockchain, etc. contextualized for different industries.
in the way best done by SAP, given its long history and experience of running almost two-thirds of world’s transactions. We’ll cover these topics in detail in the subsequent chapters.

1.3 Cloud-Native Applications: The New Paradigm

As discussed earlier, enterprises must look at their application strategies in a different light to maximize the value of the cloud. The cloud phenomenon is not just about moving infrastructure from dedicated, on-premise hardware to a shared infrastructure on a cloud provider’s data center; it’s about leveraging capabilities made available by the infrastructure into applications. Many applications have properties that are particularly suited for the cloud. Cloud-native applications are defined as applications that are “born” on the cloud; that is, these applications leverage the functionalities of the cloud and have characteristics that support the flexibility, agility, and scale typical of cloud-based solutions.

Thus, migrating and modernizing current systems of record and systems of engagement are opportunities to accelerate connections with customers and opportunities for improved speed, scale, and cost management.

Three sets of applications are particularly relevant as an enterprise moves to cloud:

- Lift and shift applications
  These applications are core stable systems that should move to cloud as is. Typically, an enterprise might consolidate, standardize, and/or automate their existing applications as much as possible before moving to the cloud.

- APIfy/containerize applications
  These applications are reused often, may need to scale up or scale down often, may be deployed quickly, etc. These applications can be converted into APIs and microservices or bundled in containers. These applications are described as “cloud enabled.”

- Refactor applications
  The advent of the cloud provides the opportunity to rewrite existing applications in new ways to make them run best on the cloud and leverage the best of capabilities offered by the cloud. These applications are described as “cloud native.”

Thus, cloud-native principles and containerization techniques will continue to be important in the future even for SAP landscapes.

1.3.1 What Are Cloud-Native Applications?

As discussed earlier, applications that are designed for cloud, created on cloud-based platforms, and run on cloud infrastructures are colloquially known as cloud-native applications.

Cloud-native technology provides the means for organizations to build and run scalable applications in public, private, and hybrid clouds. This is done via containers, microservices, service meshes, immutable infrastructure, and declarative APIs.

As we will discuss throughout this book, this new paradigm forces us to think differently about how we design our solutions. More ways to decompose applications beyond the three-tier model (frontend, application layer, and database layer) are emerging. SAP is now front and center of this movement, thus ensuring that SAP applications benefit from this new paradigm.

1.3.2 Microservice Architecture

Microservices are architectural styles in which large applications are made up of smaller components where each component focuses on a certain functionality. Each microservice normally performs one task, and they communicate with each other using language-neutral APIs. Microservices don’t need to know anything about the underlying implementation or architecture of other microservices.

The microservices architectural style is an evolution of service-oriented architecture (SOA). However, the focus of SOA was technical integration while microservices focus on integrating business capabilities.

The key design principles of microservices include the following:

- One task
  Microservices should focus on doing one job well. This focus allows for easier management and improved quality.

- Standardized communications
  Each microservice should communicate with other microservices through standard protocols like the REST API or through explicit communication patterns.

- Independence
  Each microservice should be able to independently and continuously integrate and continuously develop so that the overall application can add new features/updates without impacting the whole application.
Individual resiliency, high availability
Disaster recovery clustering can be performed for each microservice instead of using one common cluster. For the entire application, this difference saves resources.

The benefits of microservices are many and include the following:

Scalability
Scaling individual microservices as needed is easier than scaling an entire application.

Easier maintenance
Given its granular nature and its focus on a single task, microservices are easier to maintain than a big monolithic application. Upgrading individual microservices can occur without having to upgrade the entire application.

Microservices are typically aligned to one business function, and thus, its change management is easier than for a single monolithic solution.

Flexibility
Unlike monolithic applications, microservices can be developed using different technologies, that is, deployed using different frameworks like Docker, Cloud Foundry, etc.

Microservices are the key building blocks of modern cloud-native applications. However, you must define the right granularity to lower the risk of too much granularity, which would incur high integration and maintenance costs.

Special attention should be given to the data management of microservices. Microservices should ensure that the data lifecycle is managed in a safe and secure way, while complying with regulations (like GDPR, HIPAA, etc.) for different types of data (e.g., data at rest versus data in motion).

Microservices should be combined with a robust automated delivery system to provide speed at scale, which is required to make the most impact. DevOps practices like continuous development and integration should be practiced.

Operating microservices also need special consideration. While not the only option, automating the deployment of microservices for cloud-native applications is usually performed by leveraging containers.

Let’s look at the fundamentals of containers next.

1.3.3 Container Architecture
In a traditional model, when applications run on native hardware, a single app typically does not need the full resources of a single machine. Thus, most organizations run multiple apps on a single machine to avoid wasting resources. Thus, virtual machines (VMs) came into existence.

A VM is a software-based environment that simulates a hardware-based environment. Applications need an operating system and a set of processor cores to run, and they can run within a VM without any rearchitecture. VMs provide the isolation required between apps that share the same physical resource. In this scenario, a software component called a hypervisor provides the layer of abstraction where the VM can work independently of the hardware.

However, VMs have full operating system stacks, which make them relatively large and inefficient.

Containers are newer mechanisms that also provide isolation without needing to duplicate operating systems and system libraries. A container consists of an entire runtime environment bundled into one package. As shown in Figure 1.1, this package consists of:

- An application
- All its dependencies, libraries, and other binary files
- The configuration files that are needed to run the application

Containers utilize the host’s operating system, which saves space and reduces vulnerabilities. Thus, containers are more flexible and more cost effective than VMs.

To sum up, while VMs virtualize hardware, containers virtualize operating systems.

![Figure 1.1 Virtual Machine and Container Package](image-url)
A physical server running three virtual machines would have a hypervisor and three separate operating systems running on top of the hypervisor.

A server running three containerized applications can run a single operating system, and each container can share the operating system kernel with other containers.

Since containers need less resources than VMs, more containers can run on a physical machine than VMs. Containers are portable as well. Any platform with a container engine can run containers.

Containers are easy to manage. Container images are easy to share, download, and delete. Containers are easy to create and delete, and each container instance is easy and fast to start and stop.

While container technology is not new, its adoption had been slow because of difficulty of use. This situation changed with the advent of standardization through Docker, Rocket, Cloud Foundry, the Open Container Initiative, etc.

To further enhance the impact of containers, orchestrating and automatic management of multiple containers were required. Thus emerged container management software like Kubernetes, Docker Swarm, etc. These products help provision, manage, and scale apps. With this software, you can manage the lifecycle of containerized apps in a cluster of nodes (which are collections of machines, for example, VMs or physical machines).

Apps might need other resources to run, such as volumes, networks, etc., to store data, connect with other apps, etc. Container management software allows you to manage these requirements as well.

Kubernetes is an open source container management product started by Google and subsequently adopted by the CNCF and other industry majors like IBM, Amazon, etc. SAP has committed to Kubernetes Container Orchestrator as a key component of SAP Cloud Platform alongside Cloud Foundry-based PaaS. SAP Data Hub, a data management and integration application on SAP HANA, was one of the first Kubernetes-based services.

Containers as a service (CaaS) offerings based on Kubernetes are becoming popular and gaining traction. The industry is closely watching whether Kubernetes-based CaaS will overtake traditional PaaSs in near future.

1.4 Characteristics of Cloud Platforms and Applications

As discussed earlier, the cloud has brought with it a paradigm shift in the way enterprises can redesign their IT to support their digital transformation journey, from new ways to meet infrastructure needs to the ability to develop new functionalities quickly. The cloud in its many forms (IaaS, PaaS, SaaS, etc.) can play a key role in these processes.

As PaaS, cloud platforms provide enterprises the ability to develop differentiations, integrate with partners, bridge the physical-digital divide, and interact with customers in completely new ways. Thus, you must choose the right platform. While enterprises don’t normally get locked in to a platform as in the past, having the right governance in place to continuously evaluate the ever-evolving capabilities of the various cloud platforms or new providers against the characteristics needed for the enterprise.

1.4.1 Characteristics of Cloud Platforms

An enterprise’s expectations about a cloud-based PaaS may be numerous, and thus, evaluation criteria will be multidimensional as well.

The first criterion is the functionalities that the cloud platform offers, which can range from business functionalities to technical functionalities. Once you determine the functionalities needed by the enterprise, you would need to evaluate whether the nonfunctional capabilities of that platform are appropriate. Speed, stability, scalability, high availability, disaster recovery, maturity of offerings, privacy, security, data center footprint, and service level agreements are some aspects that require closer evaluation. While PaaS are meant to hide the complexities of the underlying infrastructure and allow interaction at a higher level of abstraction, thus freeing you up from operating and maintaining the platform, you should know the architecture of the platform to ensure your clients are aware of the impact on rest of their landscape. We recommend having a common unifying architecture in place that covers the entire landscape, across on-premise, private, and public cloud footprint along with their impact on the SaaS portfolio. Knowing about the operation model and ensuring that it meets your needs are also important.

A cloud platform will generally have the following layers:

- Intelligence
- Runtimes
- Development, integration services
- Infrastructure services
- Deployment services
- Security, compliance
Security and Compliance
Enterprises must assess the security provided by the cloud provider at each layer, including data center security; encryption and bring your own key capabilities; vulnerability assessments at all levels e.g., network to storage to container, etc.; geofencing; continuous monitoring; and reporting of controls that support regulatory compliance needs like GDPR, HIPAA, etc.

Deployment
Be sure to check the different deployment options of the platform: While most PaaSs offer a public cloud-only deployment model, some offer private cloud deployment models as well for clients who have special security or privacy needs.

Infrastructure
While you won’t directly deal with the infrastructure of a PaaS, if the PaaS offers multiple infrastructure options, you can support special needs. For example, to run SAP HANA-based applications, a PaaS would need specialized infrastructure. Similarly, workloads from legacy applications may need to run on bare metals or appropriate hardware.

Runtime
You’ll need to support the appropriate runtimes required for your applications. The platform should support as many varieties of runtimes as possible to account for future needs. Thus, the ability to handle new applications along with traditional applications is important. The ability to develop and run cloud-native applications using a microservice-based architecture managed by container-aware orchestration tools integrated to DevOps tools allows faster error-free releases, which is a key capability needed in a modern-day PaaS. Check the support of open source projects as well.

Development Platform
You’ll need to ensure that the PaaS has the right development environment, supporting the right programming languages, with the right tooling for both development and DevOps powered by right automation and intelligence. The major usage patterns are development, middleware deployment, and integration.

Intelligence Layer
Modern-day PaaSs are not only expected to facilitate development and integration but also to provide capabilities to derive maximum value out of the data through analytics. Intelligence can be easily introduced by developing capabilities involving machine learning, artificial intelligence, etc., as available.

1.4.2 Development in SAP Cloud Platform
One major initial driver of PaaS was to provide development environments on the cloud as enterprises wanted to release functionalities fast, get feedback early, and fail fast, if required. A development environment on a cloud platform allows developers to start development quickly without having to wait for the right development environment to be ready. Developers could access a variety of environments as new programming environments emerged, try them before using them in the long term, which would have been difficult and costly to determine otherwise. Moreover, platforms should provide development environments integrated with lifecycle management tools, DevOps capabilities, etc. Having elaborate toolsets for these different phases and activities of development lifecycle would be difficult to procure otherwise. The platform should also offer functionalities as services to expedite development and improve quality. For example, error handling, auditing, logs, user interface (UI) frameworks, security, etc. can be offered by the platform, which would save developers the effort to code these capabilities.

A platform should allow fine-grained control of the IaaS layer so that developers can deploy their applications on the right platform (bare metal, virtual servers, containers etc.) as required through infrastructure as code capabilities. You should be able to choose from a wide range of data centers to handle requirements for low latency and performance in addition to meeting any requirements from a compliance perspective.

PaaSs should offer multiple runtimes to allow developers flexibility. A runtime is a set of resources required to run an application. Runtimes involve buildpacks, which prepare the code to run on the cloud.

Integration is one of the most usage important patterns. A platform should offer standard integration protocols and preferably also offer prebuilt patterns for commonly used integrations. The ability to create, manage, and run APIs, along with the ability to connect to other cloud or on-premise setups to access legacy applications or other apps from partners and clients, is important.

Integration with the physical devices (sensors etc.) through IoT services, or services like blockchain, mobile, etc. are some of the common needs and expectations from a PaaS.
1.4.3 Agility in SAP Cloud Platform

As discussed earlier, a cloud platform provides unprecedented flexibility and speed in introducing new capabilities to an enterprise. Besides not needing to procure and set up the software and the hardware, a cloud platform accelerates the software development lifecycle at different phases through different means:

- **Development environment embedded with DevOps toolchains**
  A robust environment can help accelerate the release and management of software. In the past, development and operations teams worked in isolation. With DevOps, both teams work together collaboratively, resulting in faster and higher-quality releases. Platforms often have DevOps toolchains to facilitate automation, collaboration, and continuous feedback across the entire lifecycle. The Continuous Delivery (CD) toolchain offers an automated pipeline, automated builds, quality checks, unit tests, deployment, etc., all with minimal human interaction thereby speeding up the entire process significantly. These tools enable collaborative distributed development, again allowing the flexibility to apply the right skills independent of location or geography. Web-based IDEs and source control using GitHub allow for the faster movement of source code from different developers working on the same program in parallel can be merged automatically then built, tested, and deployed through continuous delivery and continuous testing practices. Most platforms provide different toolchains to automate as much of these activities as possible.

- **Common services**
  Typically, developers need to develop functionalities like monitoring, auditing, logging, security, etc. for their applications. Many platforms provide services to perform these functions and can be utilized as applications are being developed, thus reducing the time and effort to deliver while reducing defects as well. Typically, platforms offer multiple options for these functions and thus a high degree of flexibility as well. Developers just need to call these services instead of developing these functions, which increases the speed of development significantly.

- **Prebuilt patterns**
  Platforms often provide libraries of prebuilt patterns, codes for frequently used functionalities, interfaces, etc., which reduces time and effort to develop these functions, thus adding to the agility offered by the platform.

- **Analytical and cognitive capabilities**
  Not only is developing apps with capabilities like natural language processing, machine learning, computer vision, etc., easier, but the capabilities offered also can be used to analyze developers, developments, errors, etc., feedback that improve delivery.

The interoperability and openness of the platform allow quick integration with other platforms, and functions will work better, with the agility required in the digital age.

1.5 Hybrid Systems

While a public cloud is the target destination for many, hybrid systems are a reality and may form the major part of a transformation journey. A hybrid approach involves a combination of an on-premise hosted environment, a private cloud dedicated to one client though hosted in a service provider’s data center, and a public cloud where infrastructure is hosted in the service provider’s data center and is shared by different clients. As you assess your workload, determine what work can be moved to the cloud, prioritized by impact. Most clients would have a hybrid system for a considerable time even though the end goal might be a fully cloud-based implementation.

1.5.1 Multicloud Environments

As discussed earlier, the cloud allows you to consume services from different cloud platforms as required. As a result, many clients have multiple clouds in their landscape. Just as hybrid is a reality, multicloud is also a reality.

Along with the flexibility to consume services from multiple providers, a multicloud environment brings it complexity in governance. You must ensure that your enterprise can handle the diversity of technologies, architectures, nonfunctional requirements, operating models, security provisions, etc. You may need to augment business processes and offer guidance about the tools, like API management and other broker services, to handle the heterogeneity in technologies.
You should have a reference architecture available for your landscape, which should encompass multicloud and hybrid cloud options.

In a multicloud environment, you would lack a central dashboard to monitor all services consumed from different sources. Different service level agreements, different engagement models, and different operating procedures will have to be managed. You should communicate to stakeholders how to handle the differences among the different providers. You’ll need to define standards for data storage across platforms, as well as for authentication, security, and logging, to maintain transactional integrity across the different platforms.

While some challenges are inherent with multicloud systems, this scenario is becoming more common.

1.5.2 Multitechnology

Just as a hybrid, multicloud system is a reality, so is the existence of different technologies on the different cloud environments that your enterprise may adopt.

While this situation presents great opportunity and flexibility, you’ll need to be equipped to handle multiple technologies. You’ll need the skills to use the technologies, which may require a workforce transformation or may require leveraging other service providers. As in a multicloud system, you’ll need strategies for handling heterogeneity while maximizing the uniqueness that each individual technology offers. Thus, customers often conduct pilots to understand the strengths and weaknesses of the various technologies and recommend usage/nonusage on a larger scale. You would need to assess the maturity of the technology, along with its future roadmap. Introducing a new technology without sufficient confidence on its longevity and future fit to the enterprise’s mission brings more harm than good.

1.6 Summary

The latest technologies can help a company completely transform itself into a client-centric, employee-friendly, partner-powered enterprise. This digital transformation journey is powered by the cloud, which provides unprecedented flexibility, agility, and speed. SAP Cloud Platform, SAP’s SaaS offering, can play an important role in this journey. SAP Cloud Platform is a key player in this field and plays a central role in digital transformations, not for only companies currently using SAP, but also for new SAP customers as well.
## Contents

Preface ........................................................................................................................ 19

PART I Getting Started

1 SAP Cloud Platform: Digital Transformation and Innovation ........................................ 27

1.1 Digital Enterprise: What and Why of Digital Transformation ...................................... 27

1.1.1 What Is Digital Transformation? ..................................................................... 29

1.1.2 Path to Digital Transformation ........................................................................ 30

1.2 The Role of the Cloud in Digital Transformation ................................................... 31

1.2.1 Cloud Variants ....................................................................................................... 32

1.2.2 How Cloud Supports Digital Transformation ............................................. 32

1.2.3 Key Considerations .............................................................................................. 34

1.3 Cloud-Native Applications: The New Paradigm ..................................................... 36

1.3.1 What Are Cloud-Native Applications? ........................................................... 37

1.3.2 Microservice Architecture .................................................................................. 37

1.3.3 Container Architecture ....................................................................................... 38

1.4 Characteristics of Cloud Platforms and Applications ............................................. 40

1.4.1 Characteristics of Cloud Platforms ................................................................. 41

1.4.2 Development in SAP Cloud Platform ................................................................ 43

1.4.3 Agility in SAP Cloud Platform ........................................................................... 44

1.5 Hybrid Systems ........................................................................................................... 45

1.5.1 Multicloud Environments ............................................................................ 45

1.5.2 Multitechnology ................................................................................................... 46

1.6 Summary .................................................................................................................. 46
2 Cloud Foundry and the Neo Environment 47

2.1 Introduction to the Neo Environment .................................................. 47

2.2 Introduction to Cloud Foundry .............................................................. 48
  2.2.1 Characteristics of Cloud Foundry ....................................................... 50
  2.2.2 How SAP Cloud Platform Has Adopted Cloud Foundry ...................... 52

2.3 Cloud Foundry Services Offered by SAP Cloud Platform ...................... 52

2.4 Deployment Based on Development Environment .................................. 53
  2.4.1 Deployment for the Neo Environment ................................................ 53
  2.4.2 Deployment for the Cloud Foundry Environment ............................... 55

2.5 Use Cases Based on Development Environment .................................... 57
  2.5.1 Application Development with the Neo Environment Use Cases .......... 57
  2.5.2 Application Development with the Cloud Foundry Environment Use Cases 58

2.6 Summary .............................................................................................. 58

3 Microservices ......................................................................................... 61

3.1 What Is a Microservice? ......................................................................... 61

3.2 How Does SAP Cloud Platform Enable Microservices? .......................... 63
  3.2.1 Microservice Design Considerations .................................................. 65
  3.2.2 Twelve-Factor App ............................................................................. 68

3.3 Finding the Right Microservice ............................................................. 68

3.4 Consuming Microservices ..................................................................... 69

3.5 Summary .............................................................................................. 70

PART II Application Development

4 Developing Your First Application ........................................................... 73

4.1 Setting Up the Development Environment ............................................ 73
  4.1.1 Creating an Account in SAP Cloud Platform ....................................... 73
  4.1.2 Enabling the Neo and Cloud Foundry Trial Environments .................... 75
  4.1.3 Installing the Tools and Software ......................................................... 75
  4.1.4 Setting Up Tools and Software ............................................................. 77

4.2 Developing, Deploying, and Testing an Application in the Neo Environment ................................................................. 79
  4.2.1 Application Requirements: A Simple Application ............................... 79
  4.2.2 Required Steps for Development ......................................................... 80
  4.2.3 Application Development, Deployment, and Testing ........................... 80

4.3 Developing, Deploying, and Testing an Application in the Cloud Foundry Environment ......................................................... 91
  4.3.1 Setting Up the Cloud Foundry Server Runtime in Eclipse .................... 92
  4.3.2 Deploying and Testing Your Application ............................................. 94

4.4 Summary .............................................................................................. 97

5 Backend Application Development ............................................................. 99

5.1 Java Application Development ............................................................... 99
  5.1.1 Concept of Multitarget Application Deployment .................................... 99
  5.1.2 Building a Simple MTA Application with a Java Module ....................... 100
  5.1.3 New Application Programming Model .............................................. 103
  5.1.4 Steps for Building an Application Using an Application Programming Model ................................................................. 104
  5.1.5 Developing an Application Using an Application Programming Model ................................................................. 105

5.2 Python Application Development ......................................................... 115
  5.2.1 Why Python? ...................................................................................... 115
  5.2.2 Required Setup .................................................................................. 115
  5.2.3 Creating a Simple Python Application in SAP Cloud Platform .............. 116
5.2.4 Creating an OData Consumer Application in Python in SAP Cloud Platform ............................................ 117
5.2.5 Consuming an SAP Cloud Platform Cloud Foundry Service in a Python Application .................................................. 121
5.2.6 Authorization and Authentication Checks ............................................................................................................ 121
5.3 ABAP in SAP Cloud Platform ........................................................................................................................................ 121
5.3.1 Benefits of and Use Case for ABAP in SAP Cloud Platform ......................................................................................... 121
5.3.2 Landscape with ABAP in SAP Cloud Platform ............................................................................................................ 122
5.3.3 ABAP RESTful Programming Model ......................................................................................................................... 123
5.3.4 Next Steps ....................................................................................................................................................................... 125
5.4 Building Extension Applications for Cloud Solutions .................................................................................................. 125
5.4.1 Required Setup in the Neo Environment for Extending SAP S/4HANA Cloud ...................................................... 126
5.4.2 Developing a Backend Extension Application for SAP S/4HANA Cloud ...................................................................... 127
5.5 Application Development using SAP HANA XS Classic .............................................................................................. 132
5.5.1 Launching SAP HANA Web-Based Development Workbench ...................................................................................... 132
5.5.2 Creating Backend Objects using SAP Web IDE ............................................................................................................ 133
5.5.3 Exposing Calculation View as OData Service ................................................................................................................ 137
5.5.4 Consuming OData Service Definition in SAPUI5 .......................................................................................................... 140
5.5.5 Testing the Application ..................................................................................................................................................... 142
5.6 Application Development using SAP HANA XS, Advanced Model ........................................................................... 143
5.6.1 Configuring in SAP Web IDE ........................................................................................................................................ 143
5.6.2 Creating a Multi-Target Application (MTA) .................................................................................................................. 145
5.6.3 Creating SAP HANA Database Module ........................................................................................................................ 146
5.6.4 Creating Backend Objects .............................................................................................................................................. 148
5.6.5 Creating a Node.js Module ............................................................................................................................................... 150
5.6.6 Creating the Application Instance for the JS Application .......................................................................................... 152
5.7 Migrating Applications from the Neo Environment to Cloud Foundry ........................................................................ 153
5.7.1 Preparing XS Classic Artifacts for Migration ................................................................................................................ 154
5.7.2 Preparing the Source System for Migration ................................................................................................................ 154
5.7.3 Steps for Migration of XS Classic to XS Advanced Application .................................................................................. 155
5.7.4 Migration Report .............................................................................................................................................................. 158
5.7.5 Deploying the Migrated Application to XS Advanced .................................................................................................. 160
5.8 Summary ........................................................................................................................................................................... 162

6 Frontend Application Development ......................................................................................................................... 163

6.1 SAP Fiori Cloud .......................................................................................................................................................... 164
  6.1.1 SAP Fiori Cloud Architecture ......................................................................................................................... 165
  6.1.2 SAP Fiori Cloud Services .......................................................................................................................................... 166

6.2 SAP Web IDE ............................................................................................................................................................ 167
  6.2.1 SAP Web IDE Key Features ........................................................................................................................................ 168
  6.2.2 SAP Web IDE Full Stack Key Features ................................................................................................................... 169
  6.2.3 Customizing SAP Fiori Launchpad Using SAP Web IDE Full Stack on SAP Cloud Platform ........................................ 170

6.3 SAP Build .................................................................................................................................................................. 175
  6.3.1 SAP Build Key Features ............................................................................................................................................... 176
  6.3.2 Creating and Sharing a Prototype Using the SAP Build Plugin on SAP Cloud Platform ........................................ 177

6.4 SAPUI5 ..................................................................................................................................................................... 186
  6.4.1 SAPUI5 Key Features ............................................................................................................................................... 186
  6.4.2 SAPUI5 and Model-View-Controller ......................................................................................................................... 187
  6.4.3 SAPUI5 Control Libraries ........................................................................................................................................ 187
  6.4.4 Developing an SAPUI5 Application Using SAP Web IDE Full Stack Layout Editor and Storyboard on SAP Cloud Platform ......................................................................................................................... 188

6.5 Summary .................................................................................................................................................................. 196

PART III  Operating Applications ................................................................................................................................. 199

7 Application Lifecycle Management ............................................................................................................................. 199

7.1 Git and Its Terminology ............................................................................................................................................... 199

7.2 Continuous Integration and Continuous Delivery ..................................................................................................... 201
  7.2.1 Continuous Integration Best Practices for Java Web Landscapes ............................................................................... 203
  7.2.2 Continuous Integration Best Practices for SAP HANA XS Landscape ........................................................................ 203
  7.2.3 Continuous Integration Best Practices for SAP HANA Extended Application Services, Advanced Model, Landscape ......................................................................................................................... 204
8 Application Security

8.1 SAP Cloud Platform Authentication Options ............................................................. 217
  8.1.1 Identity Authentication Service ............................................................................. 218
  8.1.2 Tenant ................................................................................................................. 218
  8.1.3 Integration between SAP Cloud Platform and Identity Providers: Trust Settings .... 218
  8.1.4 Solution Overview for User Authentication via SAP Cloud Platform User Store .... 219
  8.1.5 Solution Overview for User Authentication Using Corporate User Store ........ 220
  8.1.6 Solution Overview for User Authentication Using a Bring Your Own Identity Provider Scenario .............................................................................................................. 221
8.2 SAP Cloud Platform Single Sign-On ......................................................................... 223
  8.2.1 Application-to-Application Single Sign-On ......................................................... 223
  8.2.2 SAP Assertion SSO ............................................................................................. 223
  8.2.3 Principal Propagation ......................................................................................... 224
8.3 Access Management in the Cloud Foundry Environment ........................................ 225
  8.3.1 User Account and Authentication Service ............................................................ 226
  8.3.2 Web Access ........................................................................................................... 226
8.4 Authorizations for Applications on SAP Cloud Platform in the Cloud Foundry Environment ................................................................................................................................. 228
  8.4.1 Access Control ...................................................................................................... 229
  8.4.2 Security Artifacts for Authorizations ................................................................. 229
  8.4.3 Authorization Enforcement .................................................................................. 230
  8.4.4 Scopes, Role Templates, and Attributes .............................................................. 230
8.5 Web Security for XSA Applications ............................................................................ 231
  8.5.1 Cross-Origin Resource Sharing .......................................................................... 231
  8.5.2 Cross-Site Request Forgery .................................................................................. 232
  8.5.3 Clickjacking ......................................................................................................... 232
  8.5.4 Cross-Site Scripting ............................................................................................. 233
  8.5.5 Timeouts for Application User Sessions ............................................................. 233
  8.5.6 Securing Node js Applications .......................................................................... 233
  8.5.7 Java Application Security .................................................................................... 233
8.6 Summary ..................................................................................................................... 234

9 Application Monitoring .................................................................................................. 235
  9.1 Monitoring Java Applications .................................................................................... 235
  9.2 Monitoring Database Systems .................................................................................. 239
9.3 Monitoring Cloud Foundry Applications ..................................................................... 241
  9.3.1 Activating the Application Log ............................................................................. 242
  9.3.2 Visualizing the Application Log .......................................................................... 242
9.4 Monitoring APIs .......................................................................................................... 244
  9.4.1 REST APIs for Java Applications in Neo ............................................................. 244
  9.4.2 REST APIs for Database Systems and SAP HANA XS ..................................... 247
9.5 Summary ..................................................................................................................... 247

10 DevOps Implementation .................................................................................................. 249
  10.1 DevOps Fundamentals ............................................................................................. 250
    10.1.1 The Core Tenets of DevOps .............................................................................. 250
## 10.1 DevOps for Hybrid Architectures
10.1.1 Setting Up DevOps for Hybrid Architectures ............................................................. 270
10.1.2 DevOps Inner Circles ................................................................................................. 274
10.1.3 Implementing Tooling for DevOps ........................................................................ 275
10.1.4 Benefits of DevOps ................................................................................................. 279
10.1.5 DevOps Tools Available in SAP Cloud Platform .................................................. 285

## 10.2 DevOps for Hybrid Architectures
10.2.1 Setting Up DevOps for Hybrid Architectures ............................................................. 286
10.2.2 DevOps Inner Circles ................................................................................................. 289
10.2.3 Implementing Tooling for DevOps ........................................................................ 292
10.2.4 Benefits of DevOps ................................................................................................. 296
10.2.5 DevOps Tools Available in SAP Cloud Platform .................................................. 302

## 10.3 Summary ......................................................................................................................... 299

## 11 SAP Leonardo

### 11.1 Design Thinking
11.1.1 SAP Leonardo and Design Thinking ...................................................................... 283
11.1.2 SAP Build and Design Thinking ............................................................................ 288

### 11.2 Internet of Things
11.2.1 Internet of Things ...................................................................................................... 291
11.2.2 SAP IoT Application Enablement ........................................................................... 295
11.2.3 SAP Leonardo IoT Service Cockpit ........................................................................ 299

### 11.3 Blockchain
11.3.1 SAP Leonardo and Blockchain ................................................................................ 303
11.3.2 Blockchain Node Creation ...................................................................................... 307
11.3.3 Blockchain Application Enablement ....................................................................... 311
11.3.4 SAP HANA Integration Service .............................................................................. 315

### 11.4 Machine Learning
11.4.1 Machine Learning ...................................................................................................... 319
11.4.2 Freestyle IoT Application Development .................................................................. 323
11.4.3 Freestyle IoT Application Development .................................................................. 327

### 11.5 Analytics
11.5.1 Live Query against SAP Data .................................................................................... 331
11.5.2 Data Acquisition or Import Connection .................................................................... 335
11.5.3 Other Features of SAP Analytics Cloud .................................................................... 339

### 11.6 Big Data
11.6.1 Hadoop .................................................................................................................... 343
11.6.2 SAP Leonardo and Big Data .................................................................................... 347

### 11.7 Data Intelligence
11.7.1 Data Intelligence ......................................................................................................... 351

### 11.8 Summary ......................................................................................................................... 355

## 12 Internet of Things Service

### 12.1 Architecture
12.1.1 Architecture ............................................................................................................... 361

### 12.2 SAP Leonardo IoT Service Cockpit
12.2.1 SAP Leonardo IoT Service Cockpit ......................................................................... 365

### 12.3 Core Service APIs
12.3.1 Core Service APIs .................................................................................................... 371

### 12.4 SAP IoT Application Enablement
12.4.1 SAP IoT Application Enablement ........................................................................... 375

### 12.5 Developing IoT Solutions with SAP Leonardo
12.5.1 Device Onboarding .................................................................................................. 381
12.5.2 Posting Data to the SAP Leonardo IoT Service Cockpit ....................................... 385
12.5.3 Creating Digital Twins Using Modeler ...................................................................... 389
12.5.4 Prerequisite Configuration: Trust Setup ................................................................. 393
12.5.5 Prerequisite Configuration: Destination Setup ........................................................ 397
12.5.6 Prerequisite Configuration: Enabling Features in SAP Web IDE ............................ 401
12.5.7 Creating Freestyle IoT Applications ....................................................................... 405
12.5.8 Deployment ............................................................................................................... 409

### 12.6 Summary
12.6.1 Summary ................................................................................................................... 413

## 13 SAP Leonardo Machine Learning Foundation

### 13.1 SAP Leonardo Machine Learning Foundation Features
13.1.1 SAP Leonardo Machine Learning Foundation Features ........................................ 417

### 13.2 SAP Leonardo Machine Learning Foundation Services
13.2.1 SAP Leonardo Machine Learning Business Services .............................................. 421
13.2.2 SAP Leonardo Machine Learning Functional Services ......................................... 425
13.2.3 SAP Leonardo Machine Learning Functional Services: Retraining .................. 429
13.2.4 SAP API Business Hub ............................................................................................ 433

### 13.3 Developing an SAPUI5 Application Using the Scene Text Recognition Machine Learning Service
13.3.1 Prerequisites ............................................................................................................. 437
13.3.2 Maintaining the Application Resource Root ........................................................... 441

## PART IV Consuming SAP Cloud Platform Services

### 13 Internet of Things Service
13.4 Internet of Things Service ............................................................................................ 445
13.4.1 Internet of Things Service ....................................................................................... 449
13.4.2 IoT Solution Cockpit ............................................................................................... 453
13.4.3 IoT Solution Cockpit ............................................................................................... 457

### 13.5 Developing IoT Solutions with SAP Leonardo
13.5.1 Device Onboarding .................................................................................................. 463
13.5.2 Posting Data to the SAP Leonardo IoT Service Cockpit ....................................... 467
13.5.3 Creating Digital Twins Using Modeler ...................................................................... 471
13.5.4 Prerequisite Configuration: Trust Setup ................................................................. 475
13.5.5 Prerequisite Configuration: Destination Setup ........................................................ 479
13.5.6 Prerequisite Configuration: Enabling Features in SAP Web IDE ............................ 483
13.5.7 Creating Freestyle IoT Applications ....................................................................... 487
13.5.8 Deployment ............................................................................................................... 491

### 13.6 Summary
13.6.1 Summary ................................................................................................................... 495

### 13.7 SAP Leonardo Machine Learning Foundation
13.7.1 SAP Leonardo Machine Learning Foundation Features ........................................ 497
13.7.2 SAP Leonardo Machine Learning Foundation Services ......................................... 501
13.7.3 SAP Leonardo Machine Learning Functional Services ......................................... 505
13.7.4 SAP Leonardo Machine Learning Functional Services: Retraining .................. 509
13.7.5 SAP API Business Hub ............................................................................................ 513

### 13.8 Developing an SAPUI5 Application Using the Scene Text Recognition Machine Learning Service
13.8.1 Prerequisites ............................................................................................................. 517
13.8.2 Maintaining the Application Resource Root ........................................................... 521

### 13.9 Summary
13.9.1 Summary ................................................................................................................... 525
PART V  Case Studies

15  Adopting SAP Cloud Platform

15.1 Use Case 1: Supplier Relationships ................................. 453
15.2 Use Case 2: Employee Rewards ........................................ 455
15.3 Use Case 3: Temperature-Sensitive Products .................. 456
15.4 Key Challenges and Lessons Learned
   15.4.1 Challenges Faced .................................................. 460
   15.4.2 Skill Development ................................................. 464
15.5 Summary ...................................................................... 465

16  Conclusion and Roadmap

16.1 The Cloud ....................................................................... 467
16.2 Platform as a Service ....................................................... 469
16.3 Neo and SAP Cloud Platform ............................................ 470
16.4 Microservices .................................................................. 471
16.5 Application Monitoring ...................................................... 473
16.6 Internet of Things ............................................................... 474
16.7 Roadmap ......................................................................... 474
   16.7.1 SAP Cloud Platform ................................................. 475
   16.7.2 Neo and Cloud Foundry Environments ...................... 475
   16.7.3 SAP Cloud Platform, Consumption-Based Business Model 476
   16.7.4 SAP Cloud Platform Services .................................. 477

The Authors ........................................................................ 481
Index .................................................................................. 483
Index

A

ABAP ........................................................ 121, 259, 478
landscape .................................................. 122
RESTful programming model .................. 123
ABAP development tools ......................... 199
ABAP frontend server landscape .............. 205
ABAP Unit ................................................... 261, 269
Access management ................................ 225
Adoption ..................................................... 453
Agile methodology ...................................... 250
Agility .......................................................... 44
Amazon Web Services ................................ 273
Analysis path framework .......................... 479
Analytics ...................................................... 286, 477
Apache Groovy .......................................... 264
APIs management ........................................ 68
monitoring .................................................. 244
REST ......................................................... 244, 247
Application access file .............................. 138
Application autoscaler .............................. 247
Application descriptor file ....................... 137
Application development ....................... 105, 132
python ......................................................... 115
SAP HANA XSA ........................................... 143
Application lifecycle management .......... 199
Application log ............................................ 242
Application monitoring ......................... 235, 473
Application programming interfaces ........ 35
Application programming model ............ 99, 105, 123
Application software as a service ............ 468
Application testing .................................... 142
Atlassian ....................................................... 200
Attributes ...................................................... 230
Authentication ............................................. 121, 225
Authorization .............................................. 121, 225
security artifacts ........................................ 229
Authorization and Trust Management service ........................................ 217
Automation .................................................. 250, 278
Autoprovisioning ...................................... 258

B

Backend objects ............................................ 133, 148
Big data ......................................................... 53, 289, 477
Big data platform ....................................... 291
Bitcoin ......................................................... 415
Blockchain .................................................. 34, 274, 280, 415, 453
application enablement ................................ 282
channel ....................................................... 424
developer node .......................................... 420
frameworks ............................................... 417
multichain node ......................................... 282
networks .................................................... 417
node creation .............................................. 282
SAP S/4HANA integration .......................... 448
Bring Your Own Identity Provider (BYOI) ... 218
Buildpacks .................................................. 43
Business application .................................. 105, 128
Business Process Change Analyzer (BPCA) ........................................ 259
Business services ....................................... 477
BYOI .......................................................... 221

C

Calculation view .......................................... 136–137, 149
CDS annotations ........................................ 110
CDS data model .......................................... 106
CDS views .................................................... 57
Chaincode ...................................................... 419
deploying ..................................................... 444
development tools ....................................... 428
Swagger test environment ....................... 446
testing .......................................................... 446
writing .......................................................... 434
Chaincode application ................................ 435
Chaincode interface .................................... 434
ChaincodeStubInterface ............................ 434
Change Request Management (ChaRM) .... 263
Clickjacking .................................................. 232
Cloud .............................................................. 31
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>JUnit</td>
<td>261</td>
</tr>
<tr>
<td>JUnit for Java</td>
<td>269</td>
</tr>
<tr>
<td>JWT</td>
<td>227</td>
</tr>
<tr>
<td>Kafka Clusters</td>
<td>419</td>
</tr>
<tr>
<td>Kerberos authentication</td>
<td>218</td>
</tr>
<tr>
<td>Kibana</td>
<td>242</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>40</td>
</tr>
<tr>
<td>container orchestrator</td>
<td>40</td>
</tr>
<tr>
<td>LDAP system</td>
<td>220</td>
</tr>
<tr>
<td>List report</td>
<td>105</td>
</tr>
<tr>
<td>Live query</td>
<td>287</td>
</tr>
<tr>
<td>Local Java server</td>
<td>86</td>
</tr>
<tr>
<td>Logstash</td>
<td>242</td>
</tr>
<tr>
<td>Long-running processes</td>
<td>65</td>
</tr>
<tr>
<td>Machine learning</td>
<td>284, 291, 453, 478</td>
</tr>
<tr>
<td>business services</td>
<td>284</td>
</tr>
<tr>
<td>functional services</td>
<td>284</td>
</tr>
<tr>
<td>MapReduce</td>
<td>289</td>
</tr>
<tr>
<td>MaxDB</td>
<td>47</td>
</tr>
<tr>
<td>Mendix</td>
<td>478</td>
</tr>
<tr>
<td>Microservices</td>
<td>35, 37, 61, 99, 471</td>
</tr>
<tr>
<td>architecture</td>
<td>37</td>
</tr>
<tr>
<td>design</td>
<td>65</td>
</tr>
<tr>
<td>Microsoft Azure</td>
<td>273</td>
</tr>
<tr>
<td>authentication</td>
<td>217</td>
</tr>
<tr>
<td>Migration assistant</td>
<td>153</td>
</tr>
<tr>
<td>Migration report</td>
<td>158</td>
</tr>
<tr>
<td>Minimum viable product (MVP)</td>
<td>256</td>
</tr>
<tr>
<td>Mobile service</td>
<td>478</td>
</tr>
<tr>
<td>Model</td>
<td>288</td>
</tr>
<tr>
<td>Model retraining</td>
<td>374</td>
</tr>
<tr>
<td>MongoDB</td>
<td>52, 477</td>
</tr>
<tr>
<td>Multicloud environments</td>
<td>45</td>
</tr>
<tr>
<td>Multitarget application</td>
<td>99–100, 145, 213</td>
</tr>
<tr>
<td>Multitechnology</td>
<td>46</td>
</tr>
<tr>
<td>Multitenant database containers</td>
<td>239</td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Neo</td>
<td>47, 100, 470</td>
</tr>
<tr>
<td>application migration</td>
<td>153</td>
</tr>
<tr>
<td>deployment</td>
<td>53</td>
</tr>
<tr>
<td>Neo environment</td>
<td>47, 132</td>
</tr>
<tr>
<td>deployment application</td>
<td>91</td>
</tr>
<tr>
<td>test application</td>
<td>91</td>
</tr>
<tr>
<td>trial</td>
<td>75</td>
</tr>
<tr>
<td>Nexus</td>
<td>203</td>
</tr>
<tr>
<td>Node.js</td>
<td>152</td>
</tr>
<tr>
<td>create</td>
<td>150</td>
</tr>
<tr>
<td>OAuth 2.0</td>
<td>226</td>
</tr>
<tr>
<td>security libraries</td>
<td>233</td>
</tr>
<tr>
<td>Nodes</td>
<td>419</td>
</tr>
<tr>
<td>NPM repositories</td>
<td>204</td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
<tr>
<td>OAuth 2.0</td>
<td>218</td>
</tr>
<tr>
<td>OAuth 2.0 workflow</td>
<td>226</td>
</tr>
<tr>
<td>ObjectStore</td>
<td>52, 477</td>
</tr>
<tr>
<td>OData</td>
<td>117, 453</td>
</tr>
<tr>
<td>consumer application</td>
<td>117</td>
</tr>
<tr>
<td>OData models</td>
<td>106</td>
</tr>
<tr>
<td>OData service</td>
<td>106, 130</td>
</tr>
<tr>
<td>OPAS tool</td>
<td>261</td>
</tr>
<tr>
<td>Open API YAML file</td>
<td>441</td>
</tr>
<tr>
<td>Open container initiative</td>
<td>40</td>
</tr>
<tr>
<td>Orderer node</td>
<td>419</td>
</tr>
<tr>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Packages</td>
<td>134</td>
</tr>
<tr>
<td>Peer nodes</td>
<td>419</td>
</tr>
<tr>
<td>Permissioned blockchain</td>
<td>417</td>
</tr>
<tr>
<td>Permissionless network</td>
<td>417</td>
</tr>
<tr>
<td>Persons</td>
<td>274</td>
</tr>
<tr>
<td>PlatOne IoT platform</td>
<td>295</td>
</tr>
<tr>
<td>Platform application router</td>
<td>64</td>
</tr>
<tr>
<td>Platform as a service</td>
<td>32, 468</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>52, 477</td>
</tr>
<tr>
<td>Postman</td>
<td>313</td>
</tr>
<tr>
<td>Google Chrome extension</td>
<td>314</td>
</tr>
<tr>
<td>machine learning</td>
<td>407</td>
</tr>
<tr>
<td>pretrained models</td>
<td>372</td>
</tr>
<tr>
<td>Principal propagation</td>
<td>224</td>
</tr>
<tr>
<td>Projection node</td>
<td>137</td>
</tr>
<tr>
<td>Python</td>
<td>115–116</td>
</tr>
<tr>
<td>machine learning</td>
<td>406</td>
</tr>
<tr>
<td>SAP Cloud Platform API Management</td>
<td>478</td>
</tr>
<tr>
<td>SAP Cloud Platform Cockpit</td>
<td>213</td>
</tr>
<tr>
<td>SAP Cloud Platform Connectivity service</td>
<td>220–221, 223</td>
</tr>
<tr>
<td>SAP Cloud Platform Identity Authorization</td>
<td>332</td>
</tr>
<tr>
<td>Administration Console</td>
<td>247</td>
</tr>
<tr>
<td>SAP Cloud Platform Integration</td>
<td>478</td>
</tr>
<tr>
<td>SAP Cloud Platform SDK for Neo</td>
<td>56, 178</td>
</tr>
<tr>
<td>SAP Cloud Platform server</td>
<td>89</td>
</tr>
<tr>
<td>SAP Code Inspector</td>
<td>261</td>
</tr>
<tr>
<td>SAP Conversional AI</td>
<td>286</td>
</tr>
<tr>
<td>SAP CoPilot</td>
<td>274, 277</td>
</tr>
<tr>
<td>SAP Data Hub</td>
<td>40, 290</td>
</tr>
<tr>
<td>SAP Data Network</td>
<td>291</td>
</tr>
<tr>
<td>SAP development tools for Eclipse</td>
<td>76</td>
</tr>
<tr>
<td>SAP Digital Boardroom</td>
<td>286, 477</td>
</tr>
<tr>
<td>SAP Document Center</td>
<td>477</td>
</tr>
<tr>
<td>SAP Enterprise Messaging</td>
<td>478</td>
</tr>
<tr>
<td>SAP Finance Health</td>
<td>371</td>
</tr>
<tr>
<td>SAP Fiori</td>
<td>105, 122, 202, 205, 275</td>
</tr>
<tr>
<td>SAP Fiori launchpad</td>
<td>473</td>
</tr>
<tr>
<td>SAP HANA</td>
<td>33, 47, 259, 281, 479</td>
</tr>
<tr>
<td>database explorer</td>
<td>149</td>
</tr>
<tr>
<td>database module</td>
<td>146</td>
</tr>
<tr>
<td>integration service</td>
<td>284</td>
</tr>
<tr>
<td>machine learning</td>
<td>369</td>
</tr>
<tr>
<td>web-based development workbench</td>
<td>130</td>
</tr>
<tr>
<td>SAP HANA database</td>
<td>57</td>
</tr>
<tr>
<td>SAP HANA Enterprise Cloud</td>
<td>33</td>
</tr>
<tr>
<td>SAP HANA smart data integration</td>
<td>53</td>
</tr>
<tr>
<td>SAP HANA XS</td>
<td>99, 203–204, 247</td>
</tr>
<tr>
<td>SAP HANA XS Classic</td>
<td>132</td>
</tr>
<tr>
<td>SAP HANA XSA</td>
<td>63, 99, 143, 204</td>
</tr>
<tr>
<td>SAP Hybris</td>
<td>273</td>
</tr>
<tr>
<td>SAP ID Service</td>
<td>217, 222</td>
</tr>
<tr>
<td>SAP Intelligent Financing API</td>
<td>371</td>
</tr>
<tr>
<td>SAP IoT Application Enablement</td>
<td>278, 297, 302</td>
</tr>
<tr>
<td>microservices</td>
<td>329</td>
</tr>
<tr>
<td>SAP Jam</td>
<td>473, 477</td>
</tr>
<tr>
<td>SAP Leonardo</td>
<td>273, 286, 295, 474</td>
</tr>
<tr>
<td>big data</td>
<td>290</td>
</tr>
<tr>
<td>blockchain</td>
<td>281</td>
</tr>
<tr>
<td>design thinking</td>
<td>274</td>
</tr>
<tr>
<td>Internet of Things</td>
<td>278</td>
</tr>
<tr>
<td>machine learning</td>
<td>284</td>
</tr>
</tbody>
</table>
SAP Leonardo Foundation ........................................ 278, 295, 297
SAP Leonardo IoT .................................................... 301
APIs ....................................................................... 300
Device onboarding .................................................. 306
Postman ................................................................... 313
Thing Modeler ........................................................... 303
SAP Leonardo IoT Bridge ....................................... 278, 296
SAP Leonardo IoT Edge ......................................... 278, 296
SAP Leonardo IoT Service cockpit .......................... 297
posting data ............................................................. 313
SAP Leonardo Machine Learning ............................ 53
business services .................................................... 370
BYOL .................................................................... 406
custom classifications .......................................... 374
foundation ............................................................. 367
functional services ............................................... 372
pretrained models .................................................. 372
RESTful APIs ........................................................ 368
retraining ............................................................... 374
TensorFlow ............................................................ 409
SAP Localization Hub ............................................. 479
SAP Localization Hub, tax service ....................... 477
SAP NetWeaver ...................................................... 214
SAP Predictive Analytics ........................................ 290
SAP S/4HANA .......................................................... 56, 273, 456
programming model ............................................. 123
SAP S/4HANA Cloud .............................................. 122
backend extension application ............................... 127
extension application ............................................ 125
SAP S/4HANA Cloud SDK ..................................... 128
SAP Screen Personas .............................................. 274, 276
SAP Service Ticket Intelligence ............................... 284
classification service ............................................. 570
SAP Solution Manager ............................................ 258
Business Process Change Analyzer (BPCA) ............ 259
Change Request Management (CRM) ....................... 263
Focused Build ....................................................... 257
Focused Run .......................................................... 260
Reverse Business Process Documentation (RBPO) ............ 260
Solution Document Assistant (SDoCa) ....................... 260
SAP Splash ............................................................ 472
SAP SuccessFactors ................................................. 48, 125–126, 289, 453
SAP SuccessFactors Employee Central ..................... 456
SAP Translation Hub ................................................ 477
SAP Vora .................................................................. 53
SAP Web Analytics .................................................. 472
SAP Web IDE .......................................................... 54, 100, 127–128, 143, 278, 473, 478
backdrop objects .................................................. 133
IoT ....................................................................... 337
SAP Web IDE full stack ........................................... 343
business application .............................................. 128
SAP Web IDE, multiclud version .............................. 91
SAPUI5 .............................................................. 47, 132, 205, 478
application resource root ....................................... 389
basic information .................................................. 388
configuration ........................................................ 390
frontend extension ............................................... 393
Inference Service for Scene Text ............................. 380
project created ..................................................... 388
template customization ......................................... 388
template selection ................................................ 386
Schema .................................................................. 134
SCIM industry standards ....................................... 217
Scopes .................................................................. 230
Secure socket shell ............................................... 50
Security ............................................................... 42, 473, 478
Service broker ....................................................... 65
Service definition .................................................. 128
Service provider .................................................... 217
Servlet .................................................................... 80
creation ............................................................... 84
Shift left method .................................................... 252
side-by-side extensions ......................................... 121
Single sign-on (SSO) ............................................. 217
Smart contracts ..................................................... 280, 430
Software as a service ............................................ 51
Solution Document Assistant (SDoCa) ....................... 260
Solution export wizard ........................................... 199, 213
Space scoped service brokers ................................ 51
Spring framework .................................................. 63
Stateless service ................................................... 63
Suppliers relationships ......................................... 453
Swagger test environment ........................................ 446

T
Table .................................................................... 134
TCP routing ........................................................... 52
Temperature-sensitive products ............................. 456
Template-based IoT ............................................... 304
TensorFlow ............................................................ 367, 406
deploy ................................................................. 411
inception model .................................................... 412
upload ................................................................. 409
Test-driven development ....................................... 201
Thing creation ....................................................... 463
Thing Modeler ....................................................... 303, 316
Tomcat ................................................................. 63
Toolchain ............................................................. 264
Transport management ......................................... 199
Transport Management System (TMS) ................... 263
Transportation management service ...................... 214
Travis CI .............................................................. 207
Trust settings ....................................................... 219
Twelve-factor app .................................................. 68
UAA service .......................................................... 226
Uniform resource identifier ..................................... 69
Unit testing ........................................................... 261
User Account and Authentication (UAA) ................. 64, 121, 226
User experience .................................................... 478
UX ................................................................. 69

Z
Zipkin tracing .......................................................... 51
Zookeeper Ensembles ............................................. 419
Gairik Acharya is a recognized expert in ABAP, SAP HANA, SAP S/4HANA, SAP Fiori, SAPUI5, SAP Cloud Platform, and SAP Mobility with over 18 years of IT experience. Currently, he works as a key architect at the IBM SAP S/4HANA Center of Excellence group in North America.

Govind Bajaj is an SAP business intelligence solution architect with over 15 years of IT experience in managing and delivering complex business intelligence solutions. He has worked extensively with the design and development of analytical solutions on the SAP HANA platform.

As an advisory consultant with IBM India, Avijit Dhar has over 12 years of experience in leading the design and implementation of multiple large-scale implementations in the SAP manufacturing domain and in growing IBM’s SAP Leonardo capability. Avijit was one of the top winners in the global IBM Cognitive Build Event 2016.

Dr. Anup Ghosh is an SAP chief technology officer at the IBM Services Europe and director of enterprise application for the IBM Cloud Solution Center. With over 23 years of experience in helping several Fortune 500 clients globally with IT transformations, Anup is an IBM distinguished engineer.

Asidhara Lahiri is a SAP executive architect in IBM India, working for global clients in different industries like consumer products, oil and gas, airlines, and utility industries, with complex SAP implementations. She has more than 22 years of work experience, out of which 18 are with SAP. She is also a coauthor of SAP S/4HANA: An Introduction, published by SAP PRESS.