



SAP S/4HANA and SAP S/4HANA Cloud

SAP S/4HANA® and SAP S/4HANA Cloud Extensibility for Customers and Partners

September 2018

Table of Contents

3 1. Management Summary

4 2. The Big Picture

6 3. Extensibility Options for SAP S/4HANA and SAP S/4HANA Cloud

3.1. Side-by-Side Extensibility

3.1.1. Overview of SAP Cloud Platform

3.1.2. Programming Model and SAP Cloud Platform SDK

3.1.2.1. Persistence

3.1.2.2. Connectivity and Business Logic

3.1.2.3. Front End

3.1.3. SAP Cloud Platform, ABAP Environment

3.1.3.1. Planned Availability in 2018

3.1.3.2. Limitations

3.1.4. Central Entry Point

3.1.5. Business Event Subscription

3.2. In-App Extensibility

3.2.1. User Interface Adaptation

3.2.2. Field Extensibility

3.2.3. Custom Business Objects

3.2.4. Custom OData Services

3.2.5. Forms and E-Mail Template Extensibility

3.2.6. Business Logic Extensibility

3.2.7. Lifecycle Management for Key-User Extensibility

3.3. Extensibility Scenarios – When to Use Which Option

26 4. Extensibility for Partner Solutions

27 5. Your Way to the Cloud from an On-Premise Environment with SAP S/4HANA

5.1. Classic Extensibility for SAP S/4HANA in On-Premise Environments

5.1.1. New Enhancement Options for SAP S/4HANA

5.1.1.1. Examples

5.1.1.2. Recommendations

31 6. Additional Documentation and Resources

1. Management Summary

The experience gained from various customer and partner projects has made one thing very clear: Most SAP S/4HANA® and SAP S/4HANA Cloud systems have been extended to fulfill customers' needs. This is due to the nature of enterprise business software, where best practices need to be aligned with custom practices, and standardized systems need to be integrated into an existing software landscape.

Therefore, extensibility must cover a broad spectrum of adaptation. This includes changes to software behavior that go beyond the capabilities of business configuration, data model extensions, data exposure and integration, layout changes to user interfaces (UIs) or forms and reports, and creation of new UIs and the customer's own applications.

In times of quickly changing business processes, extensibility projects need to be fast and efficient. For this reason, SAP S/4HANA Cloud supports either of the following approaches to extensibility:

- **In-app extensibility** – an approach taken directly within the software stack
- **Side-by-side extensibility** on SAP® Cloud Platform – a broader approach utilizing all the capabilities of a platform-as-a-service (PaaS) platform

The most important use cases for side-by-side extensions are that of building completely new UIs based on the SAP Fiori® user experience (UX) or integrating with other SAP applications. You can build completely new applications and business logic that run natively on SAP Cloud

Platform, the apps being loosely coupled with the ABAP® platform back end of SAP S/4HANA or other SAP products.

To support line-of-business (LOB)–driven adaptation projects, the most prominent patterns of extensibility (such as field extensions) are offered to key users – personas that have a deep business knowledge but no technical expertise. In addition to that, developers have all the development tools they need to achieve the most flexibility on SAP Cloud Platform.

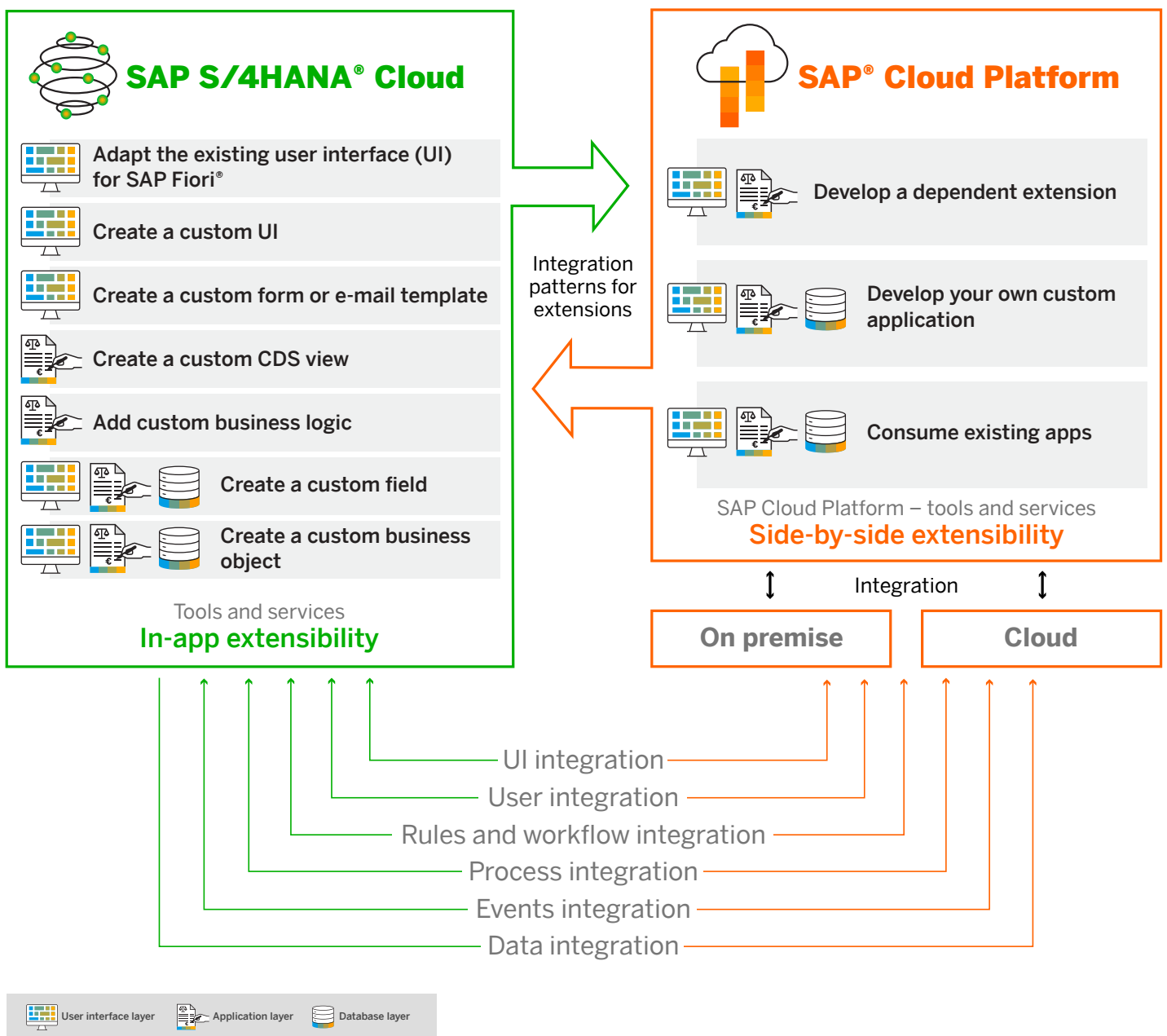
In on-premise environments with SAP S/4HANA, the in-app and side-by-side extensibility approaches are available, but in addition, **classic extensibility** – the full flexibility of extensions developed with ABAP code through the ABAP in Eclipse IDE – is also provided.

This paper describes the different extensibility features and packaging options that are either available to customers and partners now or being planned for SAP S/4HANA and SAP S/4HANA Cloud and how you can implement them.

2. The Big Picture

The extensibility options available for SAP S/4HANA and SAP S/4HANA Cloud offer various opportunities for customization. See Figure 1.

Figure 1: Extensibility Options for SAP S/4HANA and SAP S/4HANA Cloud



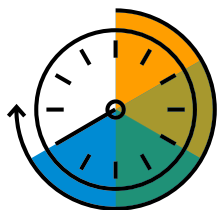
In-app extensibility focuses on adaptations created by key users, with its main features geared toward adding new fields or custom logic and adapting the standard UIs. Custom forms, reports and UIs, and custom business objects are also supported. Discovery of in-app extensibility capabilities is possible through the extensibility cockpit – a key-user app that allows developers to browse through data and process models and visualize extensibility options.

In contrast to in-app extensibility, side-by-side extensibility on SAP Cloud Platform allows developers to build their own applications, enhancing and supplementing SAP S/4HANA and SAP S/4HANA Cloud. As shown earlier in [Figure 1](#), although the coupling of side-by-side extensions is quite loose, tight integration on all architecture layers is still provided by the appropriate APIs. So, side-by-side extensibility is mainly a story of APIs and suitable programming models. All APIs available for side-by-side extensibility are documented on [SAP API Business Hub](#). The programming model (use of SAP Cloud Platform software development kit) is explained in 3.1.2.

Both in-app and side-by-side extensibility depend on the stability of the supporting APIs and data models. Therefore, SAP introduced a release, or whitelisting, concept to ensure the stability of these objects dependent upon the usage contract.

In SAP S/4HANA Cloud, only released APIs are accessible. This applies to both in-app and side-by-side extensibility. To ensure the system's integrity, especially during upgrades, and to decrease operation cost and incidents, compliance with the release concept is enforced for all extensions of SAP S/4HANA Cloud. In on-premise environments with SAP S/4HANA, the use of released APIs is not mandatory but recommended, so you can smoothly bring your custom code toward cloud stability, step-by-step.

In addition to the release concept, a deprecation concept for APIs exists. APIs that must be deprecated for any reason (for example, a new API with a more complete feature set is available) are marked as being deprecated, and a substitute API is documented. In any case, APIs will be available for a long period of time before being deprecated, and they can still be actively used for two to three releases before they are withdrawn.



Extensibility must cover a broad spectrum of adaptation, and in times of quickly changing business processes, extensibility projects need to be fast and efficient.

3. Extensibility Options for SAP S/4HANA and SAP S/4HANA Cloud

When it comes to extensibility features, both customer and partner extensions are built using the same basic elements (for example, field extensions, logic extensions, and custom business objects). However, partner solutions are far more complex than customer extensions. This is due to technical factors, such as namespace requirements and different lifecycles, as well as business factors, such as the buying process, contracts, responsibility, and maintenance. Therefore, not all the extensibility features mentioned in this paper are available for partner solutions. After reading this section, please see Section 4 for additional clarifications regarding extensibility for partners.

3.1. SIDE-BY-SIDE EXTENSIBILITY

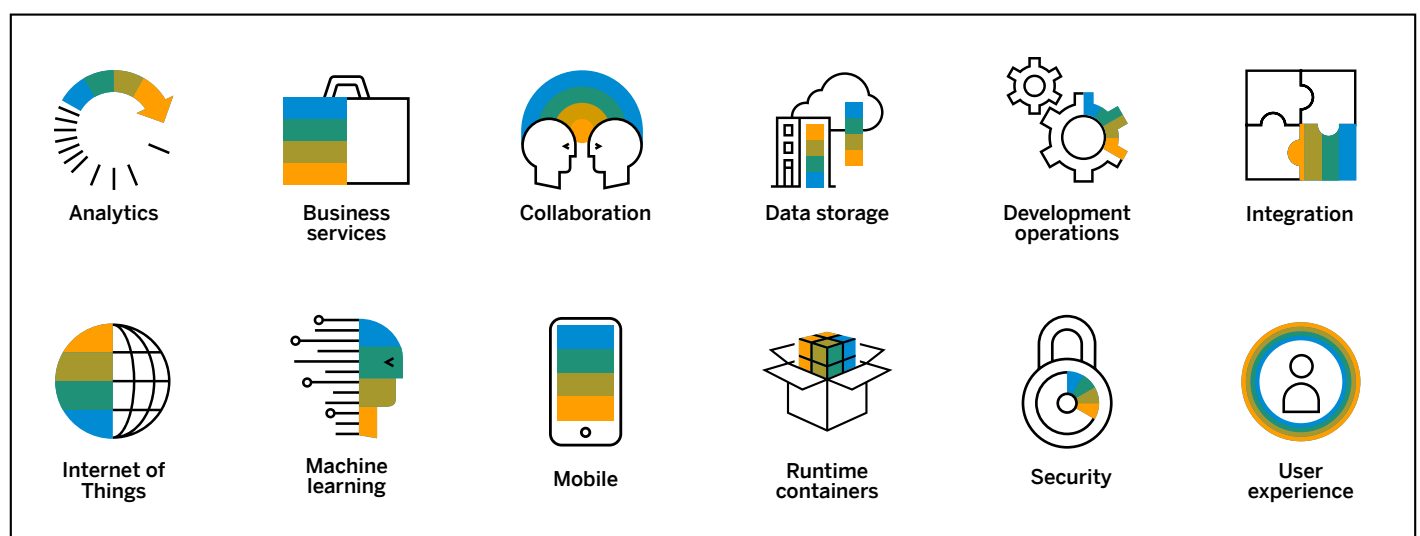
3.1.1. Overview of SAP Cloud Platform

SAP Cloud Platform is the platform-as-a-service (PaaS) offering that customers, partners, and developers can use to build side-by-side extensions by leveraging the services available in both Neo and Cloud Foundry environments.

SAP Cloud Platform comprises infrastructure services, such as cloud operations, data backup, compliance, and service-level agreements (SLAs); database services provided by the SAP HANA® business data platform, such as in-memory analytics, text search, planning, predictive, and stored procedures; and application services, such as cloud portal, security, document, administration, and development tools. Powerful integration services are available with the SAP Cloud Platform Integration service.

SAP Cloud Platform, with its broad range of extensibility services (see Figure 2), is the de facto choice for building an extension for any cloud solution from SAP. Developers can use Java, HTML5, or the native extended application services of SAP HANA, combined with the underlying open-API layer and powered by the speed of SAP HANA.

Figure 2: Extensibility Services Available on SAP Cloud Platform

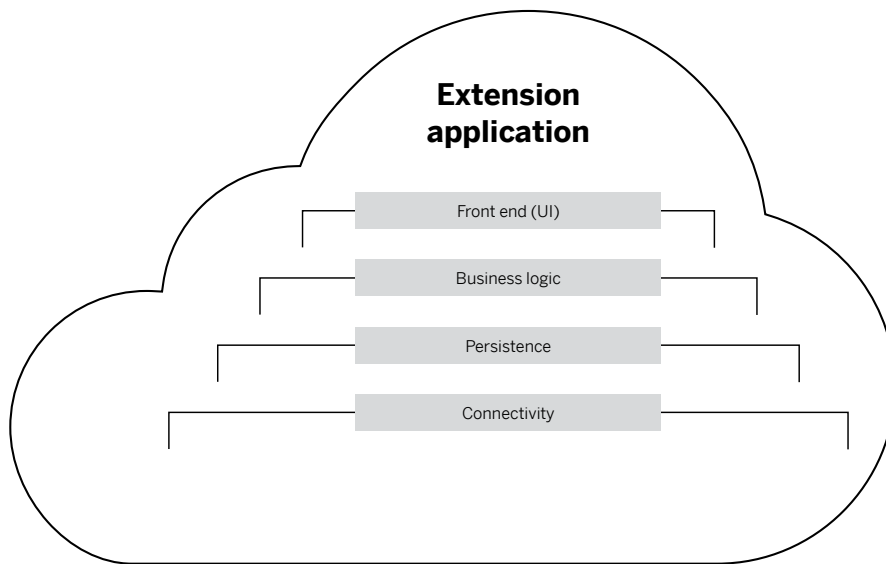


As shown in Figure 3, an extension application consists of several layers. It usually has a front-end, UI layer that is decoupled from the back end by Open Data Protocol (OData) or Representational State Transfer (REST) services. The extension application back end includes existing SAP solution services, or it can expose custom

services delivered with the extension application on SAP Cloud Platform. It comprises the following artifacts:

- Active business logic, including both the content and the security checks
- Persistence layer
- Connectivity to one or more back-end systems

Figure 3: Extension Application Layers



3.1.2. Programming Model and SAP Cloud Platform SDK

The programming model of a side-by-side extension of offerings in the SAP S/4HANA or SAP S/4HANA Cloud suite follows the approach of an independent microservice that consumes content from SAP S/4HANA Cloud through services exposed by SAP S/4HANA Cloud. An extension application, therefore, consists mainly of the following layers:

- Front end – presentation of the enriched or changed business process to the user
- Business logic – customer-specific business logic for enriching the business process to be extended
- Persistence – customer-specific persistence for data that is used exclusively by the extension application
- Connectivity – connection to SAP S/4HANA Cloud for making use of the content and applications offered by SAP S/4HANA Cloud

3.1.2.1. Persistence

For persistence, you use the service offerings that SAP Cloud Platform makes available, be it a dedicated SAP HANA database in a Neo environment or a database as a service (DBaaS) in the Cloud Foundry environment. The database content can then be consumed readily in the respective business logic layer. This is facilitated by various offerings of SAP, as described in 3.1.2.2.

In addition, in the case of DBaaS, SAP plans to enable replication of content from SAP S/4HANA Cloud into one DBaaS instance for various whitelisted business objects.

3.1.2.2. Connectivity and Business Logic

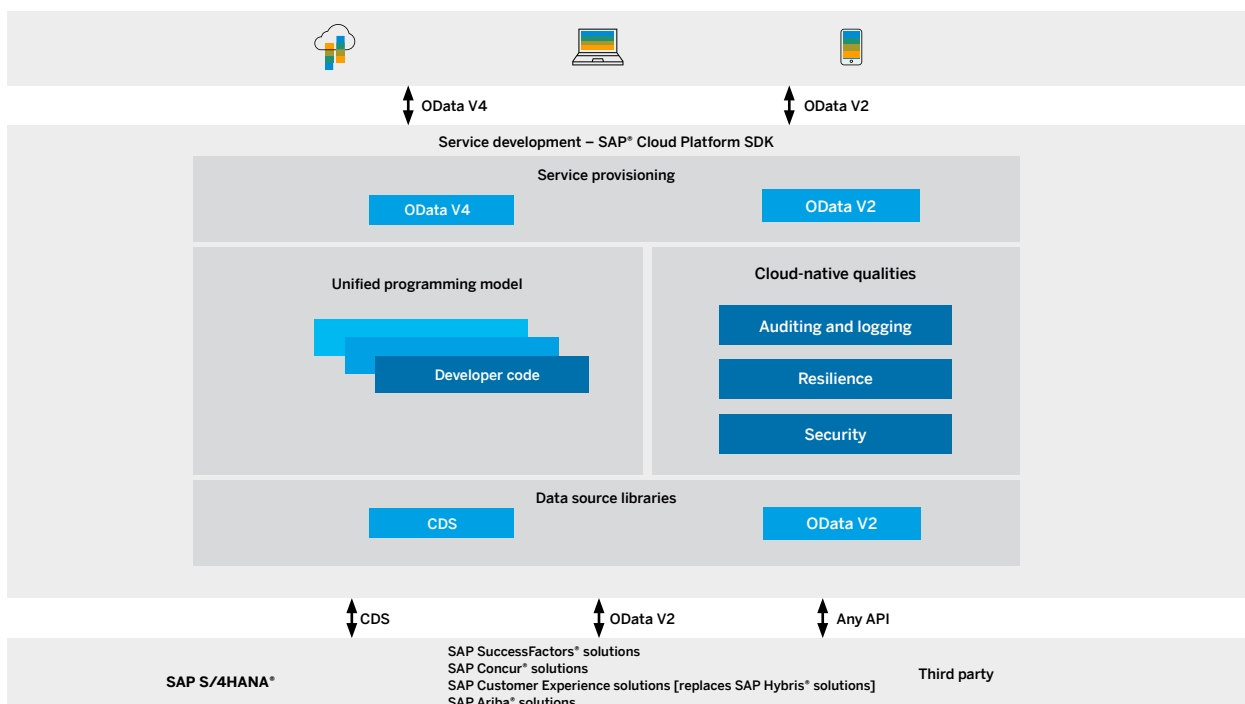
The services that are available with SAP S/4HANA Cloud are exposed through [SAP API Business Hub](#). It is SAP's goal to make the content from SAP S/4HANA Cloud

available for developers in a way that's simple to consume. For this purpose, SAP offers [SAP Cloud Platform software development kit \(SDK\)](#) as well as [SAP S/4HANA Cloud SDK](#).

SAP Cloud Platform SDK enables generic consumption and exposure of any SAP software services available, whereas SAP S/4HANA Cloud SDK offers type-safe consumption of the services of SAP S/4HANA Cloud in particular.

SAP Cloud Platform SDK (see Figure 4) is available for use in the Cloud Foundry environment of SAP Cloud Platform and facilitates development of business logic written for the Java container. It enables consumption of core data services (CDS) or OData V2 data sources and exposes those as OData V2 or OData V4 services. The exposed services enable efficient front-end consumption in a user interface based on SAP Fiori UX or in a generic interface built with SAPUI5.

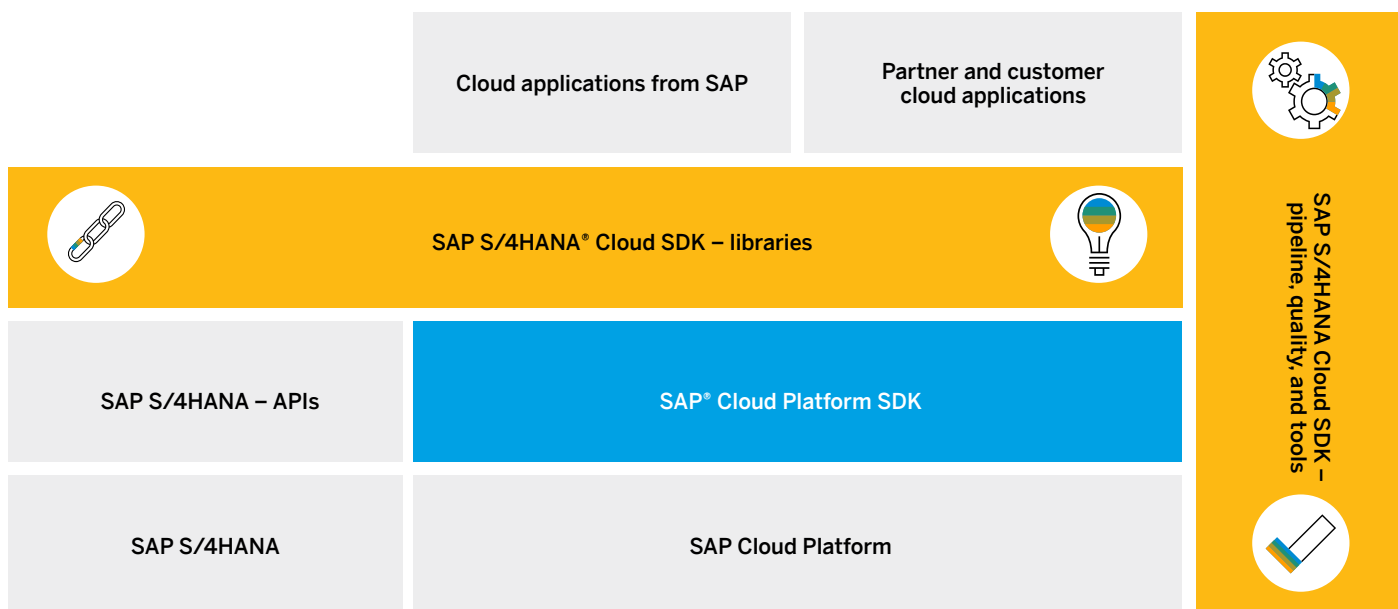
Figure 4: Architecture of SAP Cloud Platform SDK



SAP Cloud Platform SDK also offers convenient APIs for consuming either CDS artifacts from an SAP HANA database or any OData V2 service. That way, you can use the cloud application database artifacts, services from SAP S/4HANA, or other services offered through SAP API Business Hub. As a generic tool, SAP Cloud Platform SDK is not restricted to the consumption of SAP S/4HANA Cloud services alone, but it can be used to consume any OData service offered by SAP. Furthermore, it can be used to expose data through OData V2 or OData V4, which is consumed from any API such as other REST-based services.

SAP S/4HANA Cloud SDK (see Figure 5) enables access to services in SAP S/4HANA Cloud in a lightweight and type-safe manner. It is available for both the Neo and Cloud Foundry environments of SAP Cloud Platform and facilitates development for the Java container on both environments. SAP S/4HANA Cloud SDK is able to handle field extensions that key users have previously created in SAP S/4HANA or SAP S/4HANA Cloud. These extensions can be consumed alongside the standard business objects referenced by their keys either by extending the standard business objects programmatically or by using the SDK's generator to generate new classes based on the service definitions.

Figure 5: Architecture of SAP S/4HANA Cloud SDK



Among others, the following are some of the functionalities that SAP S/4HANA Cloud SDK provides to support applications:

- Connectivity – establishment and management of connections to instances of SAP S/4HANA Cloud
- Data modeling – creation of a typed data model of services in SAP S/4HANA Cloud
- Audit logging – simple access to the audit logger in SAP Cloud Platform
- Security – abstraction of security concepts for SAP Cloud Platform
- Tenant management – simple access to tenant information for multitenant applications
- Cloud quality – resilience and fault tolerance through Hystrix integration
- Persistence – multitenant awareness persistence

SAP S/4HANA Cloud SDK also supports the generation of typed access to services that are not delivered with the SDK. If business objects in SAP S/4HANA Cloud have been extended with

custom fields, or if entirely new custom business objects have been created, the SDK's generator can generate Java classes based on metadata. This allows typed access to be used in an extension application.

3.1.2.3. Front End

For an extension application's front end, an interface based on SAPUI5 or SAP Fiori UX is recommended. The use of SAP Fiori elements enables smooth, visual integration of an extension application with SAP offerings. SAPUI5 (the technology that also underlies SAP Fiori) enables efficient consumption of OData V2 or OData V4 services. Therefore, UIs based on SAPUI5 or SAP Fiori are the recommended vehicles for exposing one's business logic to the UI layer.

With SAP Web IDE, SAP offers a versatile tool that enables rapid development of SAPUI5 applications for both the Neo and the Cloud Foundry environments of SAP Cloud Platform.

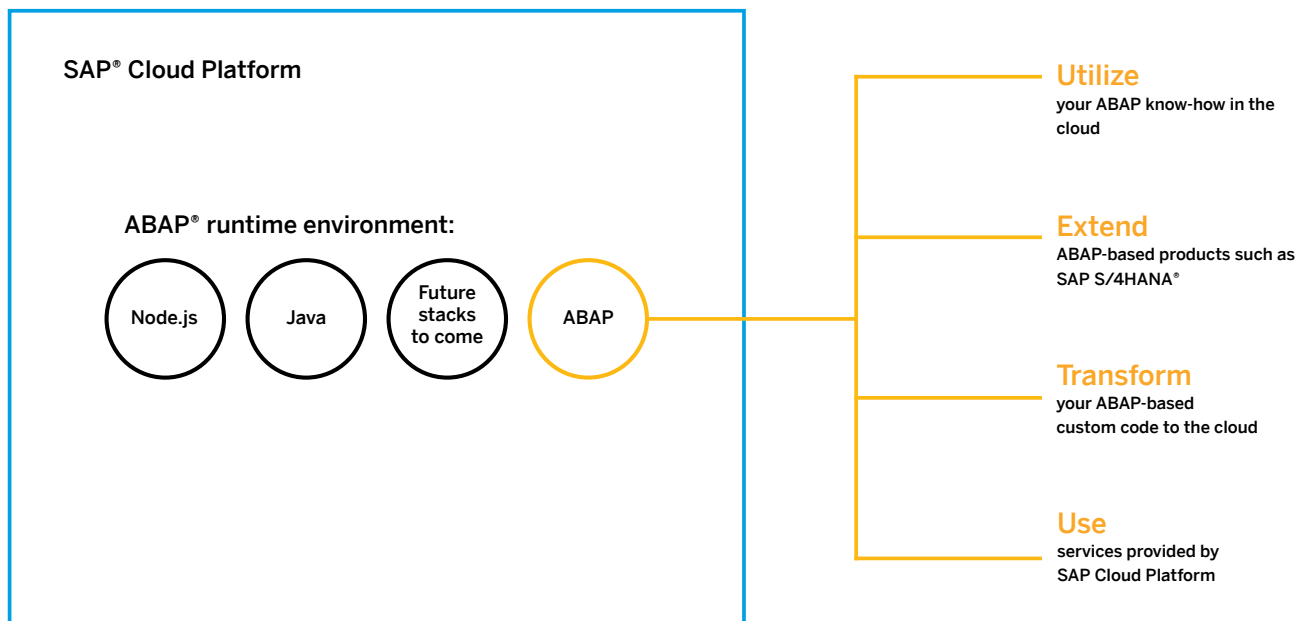
3.1.3. SAP Cloud Platform, ABAP Environment

A huge number of our customers and partners are running their mission-critical businesses with solutions based on the ABAP programming language, such as SAP ERP, SAP Business Suite, and SAP S/4HANA. Consequently, a high number of custom ABAP programs and a tremendous ABAP community has emerged over the last decades. SAP customers transitioning to SAP S/4HANA Cloud are looking for ways to leverage their investments in on-premise ABAP extensions.

With SAP Cloud Platform, ABAP environment, we plan to provide you with an additional development and runtime environment to use next to existing environments such as Java or Node.js.

While a complete rewrite of customer or partner extensions on a side-by-side, non-ABAP environment requires huge efforts and investments, you can significantly reduce the efforts required for a transition to the cloud by reusing ABAP code and skills, at least partially. For this reason, SAP plans to make ABAP available as an additional runtime environment on SAP Cloud Platform. (See Figure 6.)

Figure 6: Overview of SAP Cloud Platform, ABAP Environment



3.1.3.1. Planned Availability in 2018

SAP plans to offer a first shipment of SAP Cloud Platform, ABAP environment, that will be based on released APIs, offer a cloud-optimized ABAP language that uses a programming model based on SAP Fiori, and include the creation and consumption of CDS views.

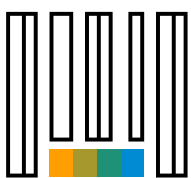
Note that a strict decoupling of customer and partner extensions from the digital core (SAP S/4HANA in the on-premise environment or SAP S/4HANA Cloud) requires a restriction to released APIs of SAP S/4HANA and the ABAP environment itself. This approach enables agile extension development and fast upgrades of SAP S/4HANA, in contrast to earlier modifications, which have often required large upgrade efforts.

Our goal is to significantly accelerate many high-volume operations of our customer or partner extensions through the use of CDS. This is a well-established and flexible methodology that uses the power of SAP HANA for data modeling and efficient data access. For example, the code push-down principle ensures that complex joins are completely executed on the database level and that only the resulting sets of queries are transferred to the application.

With SAP Cloud Platform, ABAP environment, we also intend to enable the use of non-ABAP services on SAP Cloud Platform (for example, services for document management, machine learning, and the Internet of Things), thereby increasing reuse, consistency, and development productivity.

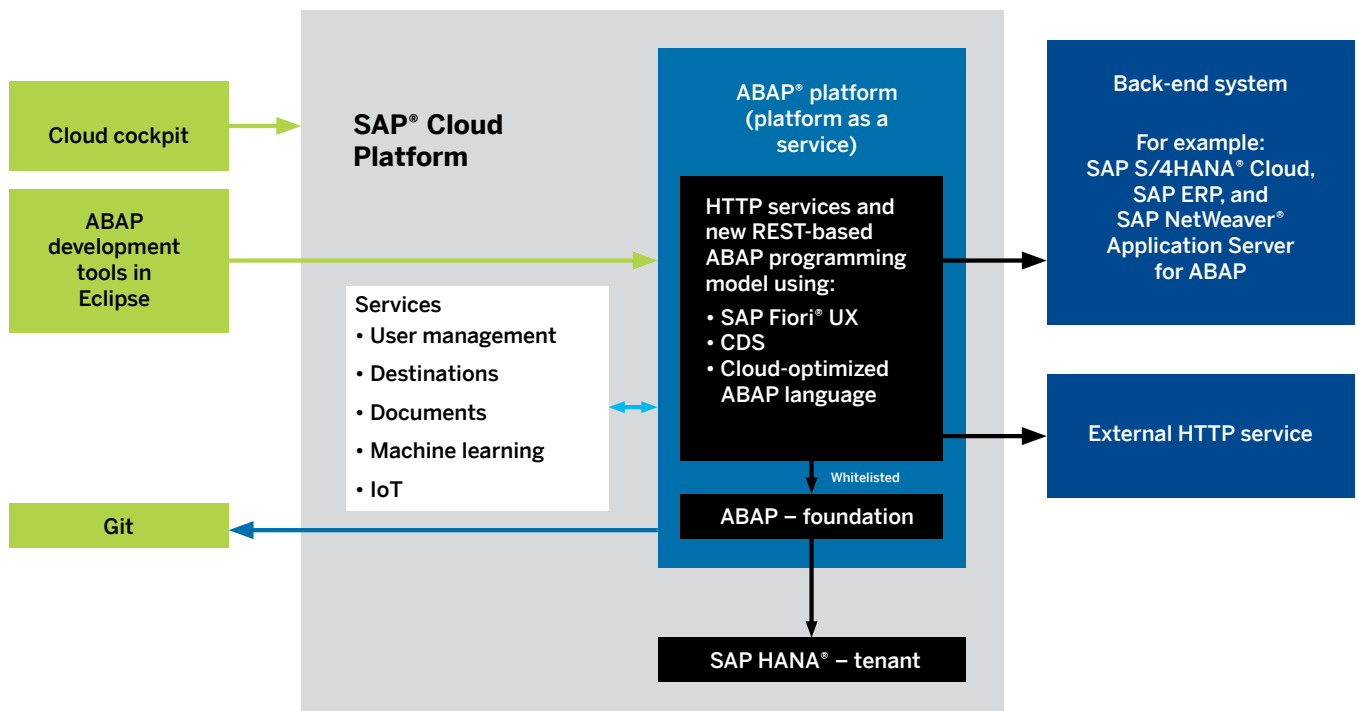
Meanwhile, ABAP development in the Eclipse IDE, which is a well-established alternative development environment for the on-premise world, besides the ABAP workbench, will be the only tool set available for ABAP development in the ABAP environment. We plan to provide native support for all ABAP development objects, which play a role in the ABAP environment, with ABAP development tools for Eclipse. Of course, for development based on SAP Fiori, SAP Web IDE will be available for use as usual.

The Git repository is another vital element that we intend to make part of SAP Cloud Platform, ABAP environment (see [Figure 7](#)). Git integration will allow ABAP objects to be stored as classes or dictionary objects in a centralized source code management system, supporting continuous integration models and transports between development, test, and productive systems.



Side-by-side extensibility, a broad approach that uses all the capabilities of SAP Cloud Platform, allows developers to build their own applications, enhancing and supplementing SAP S/4HANA and SAP S/4HANA Cloud.

Figure 7: Vital Elements of SAP Cloud Platform, ABAP Environment



3.1.3.2. Limitations

To provide a state-of-the-art cloud environment, we intend to eliminate support for legacy ABAP objects with SAP Cloud Platform, ABAP environment. This applies mainly to ABAP reports, which cannot be used together with the corresponding selection screen constructs or logical databases. In addition, we plan to have only the SAP Fiori UX serve as front-end technology; the SAP GUI interface will no longer be supported. Therefore, many of the applications based on SAP GUI will not be available in the ALV UI, or custom maintenance view, and so must be replaced in an adequate way.

In addition, we intend to use a new language version with a reduced scope from that of the ABAP language. The reduction will be achieved in two ways:

- Eliminating statements that have been obsolete for years or that are related to unsupported features (for example, “submit report,” “delete dynpro,” “perform,” “report,” and “scroll”)
- Omitting statements that could impede cloud operations such as accessing the application server’s file system (for example, “open dataset,” and “read dataset”)

Finally, to guarantee that new features planned for every quarter can be supported, we plan to allow access to the objects within the stack only for explicitly released APIs. This will provide the most meaningful feature set right from the beginning, according to our analysis of the usage of different customer and partner systems. Our goal is to increase the number of APIs gradually over the course of upcoming deliveries.

3.1.4. Central Entry Point

In a system landscape consisting of a variety of tenants and systems, the approach of using local SAP Fiori launchpads for access to these tenants and systems no longer meets all requirements. Instead, a distributed content model that is abstracted from UI technologies and product versions is a better choice. For this reason, a central entry point is on our extensibility road map, being a natural evolution of the SAP Fiori launchpad.

Our goal is to have the central entry point provide the necessary integration qualities while keeping all connected tenants and systems respected as self-contained. Specifically, no lifecycle or operations constraints will be imposed on integrated products. The central entry point will take care of provisioning tiles for launching applications, defining the navigation flows between applications, and exposing central services such as the SAP CoPilot digital assistant. We also plan to have the central entry point continue to use concepts such as intent-based navigation.

Until the central entry point is available, customers and partners have two options for adding extensions to your SAP Fiori launchpad. You can:

- Add local apps to the local launchpad, either on the ABAP or SAP Cloud Platform stack. This enables all features of the SAP Fiori launchpad, such as intent-based navigation, to be used.
- Add remote applications to the local launchpad through a URL, either on the ABAP or SAP Cloud Platform stack. This requires a greater administration effort (due to the fact that the host names are part of the URLs), and it places limitations on the features of the SAP Fiori launchpad.

3.1.5. Business Event Subscription

We offer business events as a way for customers and partners to extend processes from SAP S/4HANA on SAP Cloud Platform. Business events provide a reliable, asynchronous decoupling of the solutions in the SAP S/4HANA suite from the extension application.

You can subscribe to business events using self-service configuration in SAP S/4HANA and consume them in customer- or partner-built extension applications through the enterprise messaging and event management capabilities provided by SAP Cloud Platform.

The event management capability acts as a powerful event broker that provides all the features necessary to consume events reliably and efficiently. Client libraries are available to support this event consumption. The publication of business events must be enabled in SAP S/4HANA by configuring the integration scenario “Enterprise Event Integration.”

Currently, support of payload (for example, additional data transferred with the events beyond basic information, such as object type, event, object ID, and time stamp) as well as the number of event types is limited. Event delivery is available for all key objects and for custom business objects.

3.2. IN-APP EXTENSIBILITY

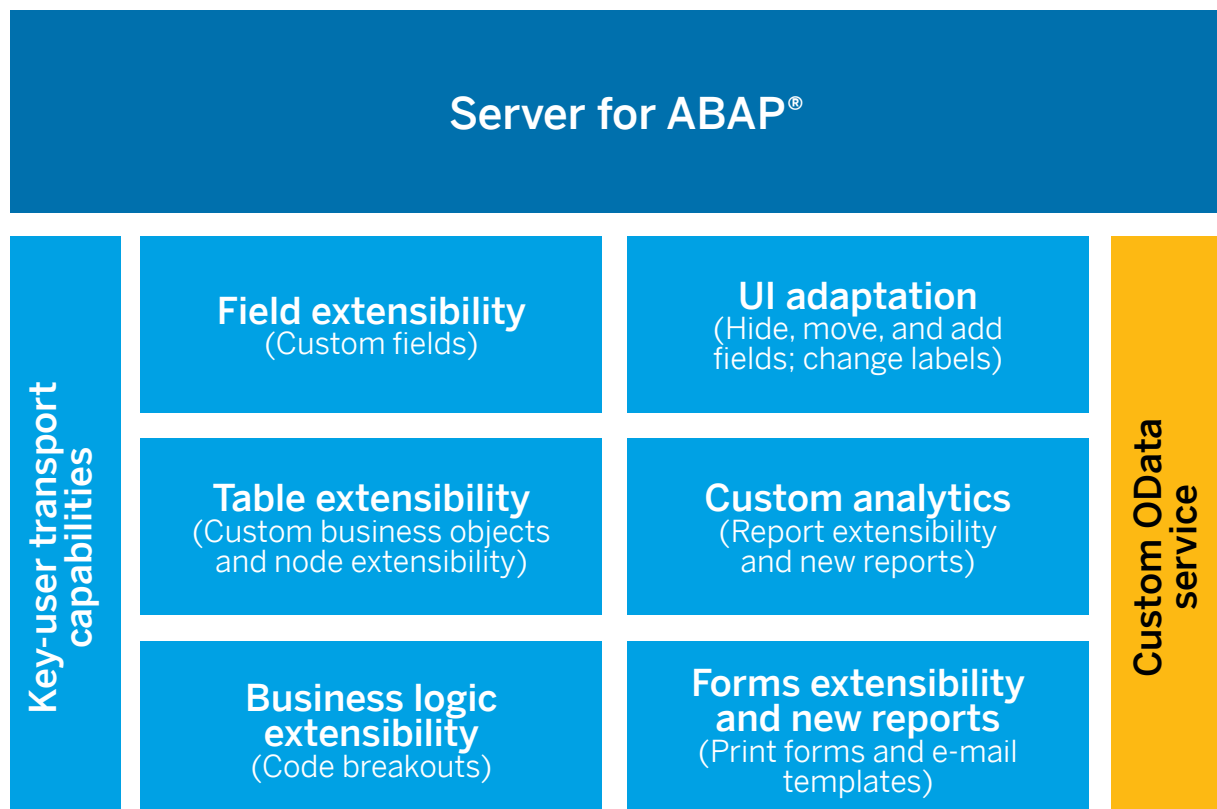
In-app, key-user extensibility tools for SAP S/4HANA and SAP S/4HANA Cloud, as well as for other LoB solutions such as the SAP Marketing solution, follow the same principles as do the leading cloud services. These include, for example, the cloud services provided with SAP SuccessFactors® solutions, Ariba® Network, SAP Customer Experience solutions, the SAP Business ByDesign® solution, and software-as-a-service (SaaS) offerings from SAP's competitors. The tools:

- Are completely Web-based (that is, developed along the principles and technology used in created SAP Fiori apps) and targeted to key users who have a deep knowledge of the product itself but not necessarily a deep technical knowledge

- Create lifecycle-stable artifacts. In other words, SAP software updates do not depend on adoptions of the extensions by the customer or partner. This means the tools are based on public, stable APIs and extension points and technology, so that they will continue to work without requiring manual activities to be performed after the next upgrade.

Extensibility features are designed for SAP S/4HANA Cloud but are also available for on-premise environments with SAP S/4HANA and have a variety of use cases. (See Figure 8.)

Figure 8: Use Cases for In-App Extensibility



Important note for customers: Management of data in an extension scenario deviates from management of data in standard scenarios. Customers are responsible for ensuring that the data used in an extension scenario is managed in accordance with applicable legal and business requirements, such as data lifecycle requirements. The extensibility framework is currently not integrated in the privacy-by-default functionality of SAP S/4HANA. Therefore, this functionality must not be used for the processing of personal data if data protection legislation is applicable.

3.2.1. User Interface Adaptation

In the adaptation mode (also called runtime authoring), the key user can adapt the UI layout in a modification-free way as follows:

- Hide fields in a form, table, or filter; hide groups and areas
- Rename labels
- Add a field to a UI from the field repository (OData service)
- Move a form field or UI group, create new groups, and move fields into a group; combine fields into one line and split combined fields
- Define new filter and table variants

The adaptation capabilities depend on the underlying SAP Fiori UX technology and the controls used and may vary from UI to UI. The help for the extended business application provides information about the adaptability of SAP Fiori apps.

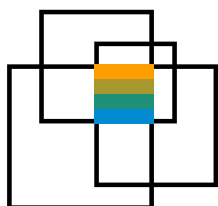
3.2.2. Field Extensibility

Field extensibility refers to the capability of adding customer-specific fields (custom fields) to a business context of an application (for example, a sales order item or a customer address). Custom field additions can be made in a one-to-one relation or by adding SAP fields that are available in tables and structures in the “lower levels” of the application to services and UIs. The tool for adding fields is key-user oriented, being launched directly from the UI of the extended application and providing guidance through a very simple extension dialog.

When several applications are part of the same business process, they can be extended together. Mapping of the custom fields in processes happens either automatically or through simple rules that can be defined by the business user.

Field extensibility enables field extensions to be reflected in the APIs of extended applications. This applies for OData services, service-oriented architecture (SOA) services, intermediate documents (IDocs), and the business application programming interface (BAPI®).

This feature is available for released business contexts, meaning an extensible table or a group of extensible tables that are extended together. As of version 1802 of SAP S/4HANA Cloud, SAP offers more than 150 extensible business contexts.



In-app extensibility, an approach taken directly within the software stack, focuses on adaptations created by key users.

3.2.3. Custom Business Objects

Table, or node, extensibility refers to the capability of adding customer-specific fields to a business context of an application in a 1:1 or 1:n relationship. (This feature is in planning and not yet available.) In contrast to field extensibility, table extensibility is intended for the creation of new customer-specific tables in the database. This would be accompanied by a set of API functionalities supporting create, read, update, and delete (CRUD) services through API classes, CDS views, and OData services.

The following provides further distinction between two types of enhancements:

- **Stand-alone custom business objects** – New, or stand-alone, custom tables that are not child tables of SAP tables are fed through a UI or data load (OData service) from other customer systems. These stand-alone tables might be used as code lists for process control or as facts or dimensions for analytical purposes. A further enhancement might be to bundle several stand-alone custom tables into a hierarchy, creating a new application with simple business logic. This functionality is available as so-called custom business objects.
- **Node extensibility** – We are planning for custom tables to be used to add fields to SAP business contexts in a 1:n relationship (for example, the hobbies of a customer) or to resolve the technical constraints of field extensibility in cases where extensions are in a 1:1 relationship. In this use case, the custom data would behave like the SAP standard data (for example, the authorization inherited from the SAP parent), and custom data would be deleted when the parent is deleted.

This use case would require that the SAP applications be prepared for this kind of extensibility, following the REST-based ABAP programming model. Thus, this feature is on our road map but not yet available.

3.2.4. Custom OData Services

SAP delivers ready-to-use APIs that are listed on [SAP API Business Hub](#). In addition, you can create new OData services that can be used by external clients as follows:

- You can expose your custom business objects as an OData service for external clients (read and write)
- You can create a new custom CDS view and expose it as an OData service for external clients (read only)

Custom CDS views can be created on top of released views of the virtual data model (VDM), which is delivered by SAP, by using a Web-based key-user tool. As of version 1802 of SAP S/4HANA Cloud, SAP offers more than 1,000 released views, including business partner, product, sales order, project, and more. With custom CDS views, you can create read-only OData services that deliver the data tailored for the needs of your external app. In custom CDS views, you can:

- Select elements from the underlying data sources (a process known as projection) and adapt their properties (for example, label, aggregation behavior, and semantics)
- Add one or more associated data sources (also referred to as “joining data sources”)
- Create calculated fields, such arithmetical calculations, case statements, and conversions
- Manage filters and parameters to restrict the result set

A preview functionality for data discovery is included in the Web-based key-user tool.

3.2.5. Forms and E-Mail Template Extensibility

Print forms are maintained through Adobe LiveCycle Designer, which is based on OData services and which might be extended in case of field or table extensions according to the user's choice. Placing a check mark on a print form that's assigned to an extended business context in the key-user extensibility tool makes the extension field available in the field catalog of the corresponding form. Navigating forward to the corresponding editor opens the administration screen for form templates, which in turn allows Adobe LiveCycle Designer to be launched.

In addition, a key user will be able to create new print forms that may be:

- Based on an existing data source (an OData service)
- Based on an extended OData service, using existing fields and associations from published CDS views
- Based on a new data source (an OData service)

The same applies for e-mail template extensibility, however, these templates are not bound against OData services but against CDS views.

This feature is available for released print and e-mail form templates. As of version 1802 of SAP S/4HANA Cloud, SAP offers more than 60 released print and e-mail form templates.



3.2.6. Business Logic Extensibility

Enhancements to the behavior of applications and processes include data validation, data calculation (for example, supplying default values), and mapping of standard and extension fields within applications and processes. To make processes more flexible, custom-specific checks (for example, on approval) or the option of removing process steps or defining additional ones might eventually be offered. Additional examples of use cases are application-domain-specific topics, such as tax calculation or price determination.

In a state-of-the-art Web-based ABAP editor, you can easily add business logic to existing enhancement options. The editor completely hides the technical details of the underlying enhancement technology (such as business add-ins, or BAdIs, from SAP) from you as a key user. It provides a simplified way of exploring APIs with key-user documentation and features such as syntax check, syntax highlighting, and code completion. Testing capabilities are also included. The editor is designed according to cloud qualities: It neither allows usage of obsolete ABAP statements nor direct database access or task handling.

In the Web-based ABAP editor, you can:

- Create simple ABAP program logic supported by syntax highlighting, code completion, syntax checking, and other state-of-the-art editor features
- Create a draft version of the program logic; test this locally; save the input values of the test as variants; publish, edit, and reset the program logic to the last published version; and delete existing custom logic
- Use released CDS views and released ABAP classes in the program logic
- Call an external Web service that runs, for example, on SAP Cloud Platform

This feature is available for released enhancement options. As of version 1802 of SAP S/4HANA Cloud, SAP offers more than 80 enhancement options.

As business users are assumed to have a different view on code than business software developers, a rule-based approach is the method of choice for many use cases. Presently, the business rules framework (also referred to as BRF+) is used to realize this capability. In a future version of SAP S/4HANA and SAP S/4HANA Cloud, we plan to integrate rule-based extensibility into the business logic extensibility.

3.2.7. Lifecycle Management for Key-User Extensibility

All key-user extensibility items follow these principles of decoupled extensions:

- All extensibility capabilities offered to customers must continue to work after an SAP software update without the need for manual work. This implies that SAP software updates do not depend on adoptions by the customer.
- Extensibility objects must never block an update of the core SAP software.

To support these core principles, the custom artifacts must comply with the following principles:

- Custom artifacts must be technically modification free; they are created as a custom object that refers to the base object.
- Custom artifacts must be technically clash free. Also, extensions must not clash with parts of the core objects that are added later. Furthermore, extensions of different extending parties must not clash. This requirement is fulfilled by applying a unique namespace that is considered in every layer of the architecture. We plan to support extensions of different extending parties (our partners and customers). For this, namespaces would require layering.
- Custom artifacts use only released, stable SAP extension points and APIs. This is enforced by object and code checks. On the SAP side, checks prevent incompatible changes to objects that are marked as released or extensible and have been delivered once. SAP plans to provide a deprecation mechanism in the future.

For business-critical applications, extensions are typically created and tested before the extension is active for all users in the production environment. Here, we distinguish between the cloud environment (SAP S/4HANA Cloud) and the on-premise environment (SAP S/4HANA).

In SAP S/4HANA Cloud, the transport of extensibility objects from the test to the production system is performed by the key user without interaction with the service provider and outside of the maintenance window of the service provider.

In SAP S/4HANA, customers have much greater freedom for development and for setting up the system landscape and quality assurance processes. Therefore, customers can manage SAP updates and customer transports for key-user extensibility with traditional transport tools (that is, ABAP change and transport system, or CTS, and all SAP and partner tools using CTS). Transports of extensions created with key-user tools can be merged with transports with custom development performed with classic ABAP development tools.

3.3. EXTENSIBILITY SCENARIOS – WHEN TO USE WHICH OPTION

Due to its simplicity and strict cloud qualities, key-user extensibility has only limited possibilities. This is especially obvious when comparing the in-app extension features with the development capabilities of SAP Cloud Platform. Nevertheless, there are strong use cases for both the in-app and side-by-side approaches to extensibility.

Knowing when to do what will always require consideration of the details of customer and partner projects, the skills of the employees, the history of existing extensions, and the customer project road map. Figure 9 provides a short overview that highlights the personas and use cases for the two options.

Figure 9: Personas and Use Cases for In-App and Side-by-Side Extensibility



Table 1 provides more details about when to use which option. Please keep in mind that you do not have to choose only one of the three options. In many cases, a combination of all options will provide the best results.

Table 1. Guidelines for Choosing Extensibility Options

Aspect	Custom development on SAP® Cloud Platform (Java, Node.js, or ABAP)	In-App Extensibility using Web-based (SAP Fiori® UX-based) key-user tools	Custom development using ABAP® development tools (ABAP workbench or ABAP in Eclipse)
Availability	Available for: <ul style="list-style-type: none"> • SAP S/4HANA® Cloud • SAP S/4HANA (an on-premise environment) • SAP Business Suite (an on-premise environment) 	Available for: <ul style="list-style-type: none"> • SAP S/4HANA Cloud • SAP S/4HANA (an on-premise environment) 	Available for: <ul style="list-style-type: none"> • SAP S/4HANA (an on-premise environment)
Integration requirements	Stand-alone application or new process step along with occasional remote usage of data: <ul style="list-style-type: none"> • Data is read less frequently from the core product (few roundtrips) through the API, or data is replicated. • Custom data is not changed together with core data (no transactional consistency required; follows BASE principle, eventual consistency is sufficient). 	<ul style="list-style-type: none"> • Close integration with the core product: • Data is frequently read from SAP S/4HANA or SAP S/4HANA Cloud (many roundtrips) • Custom data is changed together with core data (transactional consistency; ACID principle is required) 	
Scenarios	-	<ul style="list-style-type: none"> • Last-mile extensions¹ – extensions that adapt business processes for legal or industry- and company-specific best practices: • Custom fields, custom logic (BAdIs), and custom business objects • Custom CDS views, custom analytical queries, and custom forms • Custom services (OData) 	Complex enhancements (for example, complex BAdI implementations, enhancement points, and class extensions) that need deeper development skills and bigger development teams, or where no cloud-enabled, released extension points exist

1. Also known as RICEF or FRICE objects. See definitions in ricef.blogspot.de/2008/08/basics-whats-ricef.html, wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=150569211, de.wikipedia.org/wiki/RICEFW (in German but can be translated), and www.fsa.usda.gov/Internet/FSA_File/wbscm_update_0807.pdf.

	Custom development on SAP® Cloud Platform (Java, Node.js, or ABAP)	In-App Extensibility using Web-based (SAP Fiori® UX-based) key user tools	Custom development using ABAP® development tools (ABAP workbench or ABAP in Eclipse)
Scenarios	Data hub: <ul style="list-style-type: none"> • Build a data hub for collecting data across company units in a cloud system • Use analytics and other tools for data analysis 	Data model extensions of the system of records: <ul style="list-style-type: none"> • System of records is extended system of records by creating custom fields and tables • Custom data is uploaded from external systems, brought together in the SAP S/4HANA or SAP S/4HANA Cloud system and then used for reporting along with SAP standard data (for example, in embedded analytics or for segmentation in marketing solutions) 	
	Stand-alone application or new process step with occasional usage of data	Small stand-alone application or new process step in SAP S/4HANA or SAP S/4HANA Cloud with heavy usage of SAP S/4HANA or SAP S/4HANA Cloud data; can be built with custom business objects	Larger stand-alone application or new process step in SAP S/4HANA or SAP S/4HANA Cloud with heavy usage of SAP S/4HANA or SAP S/4HANA Cloud data
	Freestyle UI development, specifically native mobile UIs or consumer-grade UIs	-	-
	Loosely coupled service or microservice	-	-
	Integration hub: Build a hub to integrate with other cloud solutions	-	-

	Custom development on SAP® Cloud Platform (Java, Node.js, or ABAP)	In-App Extensibility using Web-based (SAP Fiori® UX-based) key-user tools	Custom development using ABAP® development tools (ABAP workbench or ABAP in Eclipse)
Target group (business users)²	User who do not use SAP solutions: <ul style="list-style-type: none"> • Employees: new user roles or new departments • External users: customers, prospects, or interested parties 	-	-
Target group (extension creators)	Developer: deeply skilled in programming languages (for example, Java, Node.js, and ABAP) and programming concepts	Application experts (key users and application consultants): very knowledgeable of the product's business processes, know about the data model and configuration options, and have some basic programming knowledge ³	Developer: Deeply skilled in the ABAP programming language and programming concepts
Lifecycle stability and APIs⁴	Extensions are lifecycle stable and enforced through the user of released (remote) APIs of the core product	Extensions are lifecycle stable and enforced by the usage of released (remote and local) APIs and extension options of the core product	Extensions are not lifecycle stable and the effort for adaptation during the update of the core product depends strongly on the extension technologies used
Lifecycle management	Dedicated lifecycle management	Integrated lifecycle management together with the core product (transports, monitoring, and support)	

2. Users who already use other SAP solutions (extensions to existing user roles)

3. Code breakouts are used for a well-defined purpose if a graphical or form-based UI is not suitable. Examples include formulas in custom CDS views and custom analytical queries, ABAP code for validations and calculations in ABAP language versions for key users, and the cloud.

4. See also blogs.sap.com/2017/04/12/s4hana-extensibility-how-to-find-apis-and-extension-points.

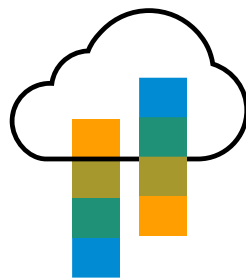


Knowing when to use which extensibility option requires consideration of the details of the customer or partner project, the skills of employees, the history of existing extensions, and the project road map.

If you opt for side-by-side extensibility, the decision about the development and runtime environment (Java, Node.js, or ABAP) still needs to be made. You might consider the following influencing factors:

- **Developer's skill set** – If you have many ABAP developers in your organization, you should consider the ABAP environment.
- **Existing software assets** – If you use many custom or partner ABAP extensions of SAP ERP, SAP Business Suite, or SAP S/4HANA, you should consider the ABAP environment (keeping in mind that the side-by-side programming model does not allow direct local access of business functions in SAP software, and existing code needs to be renovated).

- **Scenarios and use cases** – In general, ABAP-based extensions usually focus on business logic closely related to SAP standard solutions, and they are intended for employees or business partners but not for consumers. Please also consider the technical elements, such as architecture, application design, system landscapes, and lifecycle management. These should be taken into account before you make a final decision.



SAP Cloud Platform is the PaaS offering that customers, partners, and developers can use to build side-by-side extensions by leveraging services available in both Neo and Cloud Foundry environments.

4. Extensibility for Partner Solutions

As mentioned at the beginning of Section 3, because of the complexity of partner solutions, not all the extensibility features discussed in this paper are available for partner solutions.

The method of choice for partner solutions is to build them on SAP Cloud Platform. Using this side-by-side approach on a well-integrated PaaS offers the best combination of rich features, high stability, loose coupling, and the ability to build on top of several SAP products. The standard way to run partner solutions on SAP Cloud Platform is to use its multitenancy architecture. The partner acts as an SaaS provider on top of SAP Cloud Platform (your platform as a service). Thus, building on SAP Cloud Platform helps partners to get the most value from their solutions.

For some extensibility patterns such as field extensibility or custom APIs, there is no alternative to in-app extensibility (as described in section 3.3). To clarify the possibilities for partners, we place partner solutions into the following categories:

- On-behalf solutions (also known as one-offs) – The solution is built specifically for a single customer in a customer system and customer namespace.
- Template solutions:
 - The solution is built in the partner development landscape in the partner namespace.
 - It can be installed once in many customer systems and adapted to customer needs afterward.
 - Patches and corrections are performed in customer systems.

- Multicustomer solution:
 - The solution is built in the partner development landscape in the partner namespace for many customers.
 - Continuous development and corrections and patches are deployed to customer systems by a partner or SAP.
 - Various solutions for customer buying and software deployment exist.

For the on-premise environment with SAP S/4HANA, standard development tools can be used to build partner solutions of previously mentioned categories and to ship them. The [SAP S/4HANA Requirements for Partner Solutions: On-Premise Guide](#) describes the necessary criteria that needs to be fulfilled from SAP's perspective.

For SAP S/4HANA Cloud, only on-behalf solutions are supported through the use of key-user extensibility in customer systems. For template and multicustomer solutions, a fixed road map is not yet available. However, SAP has decided to offer only key-user extensibility to all three solution categories. Together with this investment, SAP plans to also build mechanisms to bundle deployment of in-app and side-by-side extensions.



The method of choice for partner solutions is to build them on SAP Cloud Platform – a side-by-side approach combining rich features, high stability, loose coupling, and the ability to build on top of SAP products.

5. Your Way to the Cloud from an On-Premise Environment with SAP S/4HANA

5.1. CLASSIC EXTENSIBILITY FOR SAP S/4HANA IN ON-PREMISE ENVIRONMENTS

For many years, customers and partners have extended and modified SAP Business Suite software. In this paper, we refer to this kind of extensibility as “classic.” In the on-premise environment with SAP S/4HANA, you still have full access to classic extensibility using development tools such as Eclipse or the ABAP workbench (SE80).

Modifications to SAP S/4HANA objects and the use of all SAP S/4HANA objects, except for “quarantined” objects (meaning objects that have been deprecated and are not directly accessible), are allowed.

Thus, the classic extensibility option combines a maximum of freedom and unlimited expressiveness with the ability to influence the process behavior of SAP S/4HANA. But the scalability, especially with respect to cloud operations, is very limited. Therefore, when modifications or the use of ABAP objects that are not on the whitelist exist, the deployment of classic extensions is limited to SAP S/4HANA.

5.1.1. New Enhancement Options for SAP S/4HANA

Over the years, SAP has developed a set of technologies for enhancing SAP code. These are:

- User and system exits (for example, the very prominent SD user exit MV45AFZZ)
- Explicit enhancement options:
 - BAdIs (for example, the “old” BAdI ME_PROCESS_PO_CUST for purchase orders)
 - Enhancement points and sections
- Implicit enhancement options (for example, exists at the end of methods, functions, and forms)
- Other application-specific enhancement frameworks that, for example, generate code exits or provide a registration and an interface that can be implemented by the extender

These technologies have their sweet spot and contribute their fair share to the success of on-premise SAP products, but only BAdIs fulfill the requirements of a save operation in a cloud SaaS solution.

Consequently, SAP provides a new set of enhancement options (based on the proven BAdI technology) for the on-premise environment with SAP S/4HANA and for SAP S/4HANA Cloud.

For SAP S/4HANA Cloud, only the new BAdIs can be used with the Web-based ABAP editor and with the restricted ABAP for key users.

In the on-premise environment with SAP S/4HANA, however, you have the following options:

- Implement the new BAdIs with the Web-based ABAP editor and the ABAP version for key users
- Implement the new BAdIs using the ABAP development tools (ABAP workbench) with the regular ABAP version or the ABAP version for key users
- Implement the old enhancement options using the ABAP development tools (ABAP workbench) with the regular ABAP version (or the ABAP version for key users)

New BAdIs for SAP S/4HANA are typically introduced in parallel with the old enhancement options, but this depends on the specific application. You must check the application-specific documentation and the SAP S/4HANA migration

guide. The following can be considered “rules of thumb,” and deviations may occur for concrete applications:

- Old enhancement options are not obsolete in SAP S/4HANA (unless the complete application module is obsolete).
- In case of a migration from SAP ERP to SAP S/4HANA, the old enhancement options typically continue to work, but the custom code must be adopted according to the migration guidelines (for example, simplification items and checks of code migrated to SAP S/4HANA).
- New BAdIs for SAP S/4HANA and old enhancement options typically run in parallel in the application code and are independent of the UI technology (SAP GUI or SAP Fiori UX, but this may also depend on the concrete application).
- The new BAdIs do not contain the complete functionality of the old enhancement option. (They are restricted to changes to the business object that allows save operations in the cloud.)

5.1.1.1. Examples

Table 2 shows two prominent examples for old and new enhancement options.

Table 2. Enhancement Option Examples

Business object	Old enhancement option	New enhancement option
Sales order	User exit MV45AFZZ	BAdIs in enhancement spot ES_SD_SLS_EXTEND (for example, sales header modification or sales document check before a save)
Purchase order	BAdI ME_PROCESS_PO_CUST	BAdIs in enhancement spot ES_MMPUR_PROCESS_PO_CLOUD (for example, purchase order header modification or final purchase order check before a save)

5.1.1.2. Recommendations

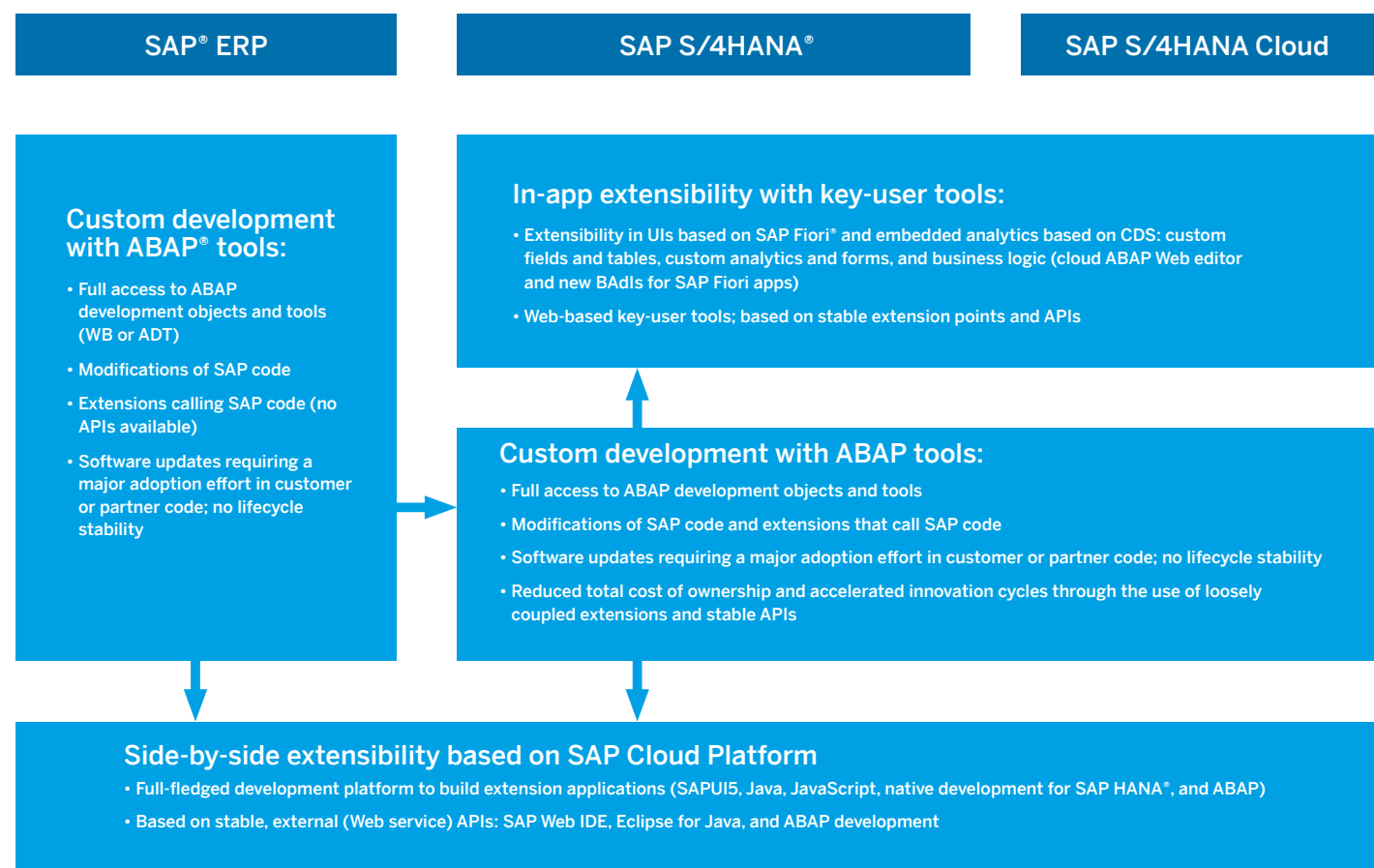
In general, if multiple options are available for your enhancement use case, choose the following, shown in order of preference, for new implementations:

- New BAdIs for SAP S/4HANA
- Explicit enhancement options, user or system exits, or application-specific enhancement frameworks
- Explicit enhancement options
- Modifications of SAP code
- Clone and adoption of SAP code

You might consider migrating existing code to new BAdIs for SAP S/4HANA.

Considering these recommendations and the recommendations provided in 3.3 for when to use which option, [Figure 10](#) summarizes our advice to on-premise customers, especially in cases where customers are migrating from SAP ERP.

Figure 10: Recommendation Summary for On-Premise Customers



6. Additional Documentation and Resources

The intention of this paper is simply to provide an overview of existing extensibility options or those that are on the road map.

You can find additional details here:

- [Blog posts on extensibility topics](#)
- [ABAP extensibility community on SAP Community](#), featuring blog posts and tutorials on in-app extensions
- [Developer tutorials on ABAP extensibility](#)
- [Developer topics on SAP S/4HANA Cloud SDK](#)
- SAP Community blog post, “[Overview of ABAP in SAP Cloud Platform](#)”
- [SAP API Business Hub](#)

An [extensibility explorer](#) that provides a centralized approach for locating extensibility documentation is available. It is our goal to have the explorer offer an overview of extensibility best practices and concrete enhancement points without requiring access to an SAP S/4HANA Cloud system. Also, we are planning for the explorer to include document enhancement options for SAP S/4HANA.



Follow us



www.sap.com/contactsap

58725enUS (18/09)

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

See <https://www.sap.com/copyright> for additional trademark information and notices.

THE BEST RUN

