

Clean core extensibility for SAP S/4HANA Cloud

January 2025, version 2.0

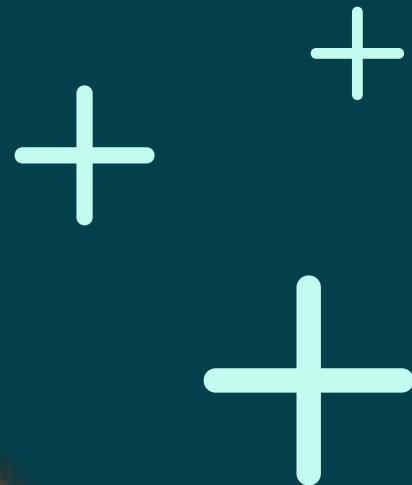


Table of contents

1	Introduction	3
2	The clean core strategy	4
2.1	Clean core principles: guidelines and benefits of extensibility	5
3	Extensibility of SAP S/4HANA Cloud	7
3.1	The extensibility model	7
3.2	Extensibility options: when to use what?	8
3.3	The public cloud extensibility model	16
3.4	The private cloud extensibility model	17
3.5	Launching a clean core certification program for partner applications	19
3.6	APIs, events, and extension points	20
4	Extend SAP S/4HANA with SAP Build	21
5	Getting started	23
6	Summary	25
7	Appendix	26

1 Introduction

Software is designed with a scope in mind. While the breadth of scope covered by SAP applications is an important factor for the adoption of SAP solutions, these solutions cannot cover the distinct need of every customer to differentiate with a competitive advantage. Therefore, SAP solutions can be extended to fit customers' unique needs. SAP introduced the revolutionary approach of installing a system with standard functionality (based on decades of best practices) along with the ability to extend that functionality in a cost-effective manner. This balanced approach has become one of the most important factors in the phenomenal success of the SAP ERP application.

To accommodate varying business needs and preferences while adhering to cloud standards, SAP introduced two different editions of SAP S/4HANA Cloud: SAP S/4HANA Cloud Public Edition and SAP S/4HANA Cloud Private Edition. For SAP S/4HANA Cloud Public Edition, as a SaaS solution, SAP makes sure the system is up to date and upgraded to the latest release for customers. This allows them to leverage a standardized and quick-to-deploy ERP solution that can be extended while keeping the core of the system unchanged, that is, helping ensure smooth upgrade cycles by not changing its base code line.¹

While upgrades in SAP S/4HANA Cloud Public Edition are managed by SAP, SAP S/4HANA Cloud Private Edition follows a different approach. It allows for more flexibility and control. SAP takes care of system operations, but upgrades are customer-managed projects that follow the customer's schedule. The higher degree of flexibility and control can lead to costly and time-consuming upgrade projects. One reason for this is that extensibility options are much more flexible – they can be decoupled from but also be modifying the core. The latter might slow

down upgrades or might even break them so that additional adjustment effort is required. Therefore, the ultimate goal of every extension project should be to keep the core of a system clean by minimizing these disturbing factors while adhering to cloud standards.

The clean core strategy and the extensibility model described in this paper support this objective: enabling customers of SAP S/4HANA Cloud² to extend where needed but still allowing for smooth system upgrades and handling of system extensions. Ultimately, the main goal is to keep the core of the ERP system as close to the standard as possible while extending the functional scope where needed.

This paper gives details on the clean core strategy in [Chapter 2](#), followed by an in-depth description of the extensibility of SAP S/4HANA Cloud in [Chapter 3](#). The extensibility model for both editions of SAP S/4HANA Cloud, as well as the possible options, are outlined with reasoning for when to use which. [Chapter 4](#) highlights the evolution of SAP Build solutions into a unified and AI-enabled clean-core extensibility solution optimized both for cloud ERP and other cloud solutions from SAP. [Chapter 5](#) gives guidance on how to get started on the journey toward achieving and keeping a clean core, followed by a summary in [Chapter 6](#) and a collection of relevant links in [Chapter 7](#).

1. Public cloud solutions from SAP for lines of business, such as SAP SuccessFactors, SAP Concur, and SAP Fieldglass solutions, offer similar extension options adhering to the public cloud principle of keeping the core unchanged. What holds for SAP S/4HANA Public Cloud Edition and is explained in Section 3.3 similarly holds for these solutions, that is, offering extension options on the system stack as well as being decoupled from it with SAP Business Technology Platform.
2. Including both SAP S/4HANA Cloud Private Edition and SAP S/4HANA Cloud Public Edition.

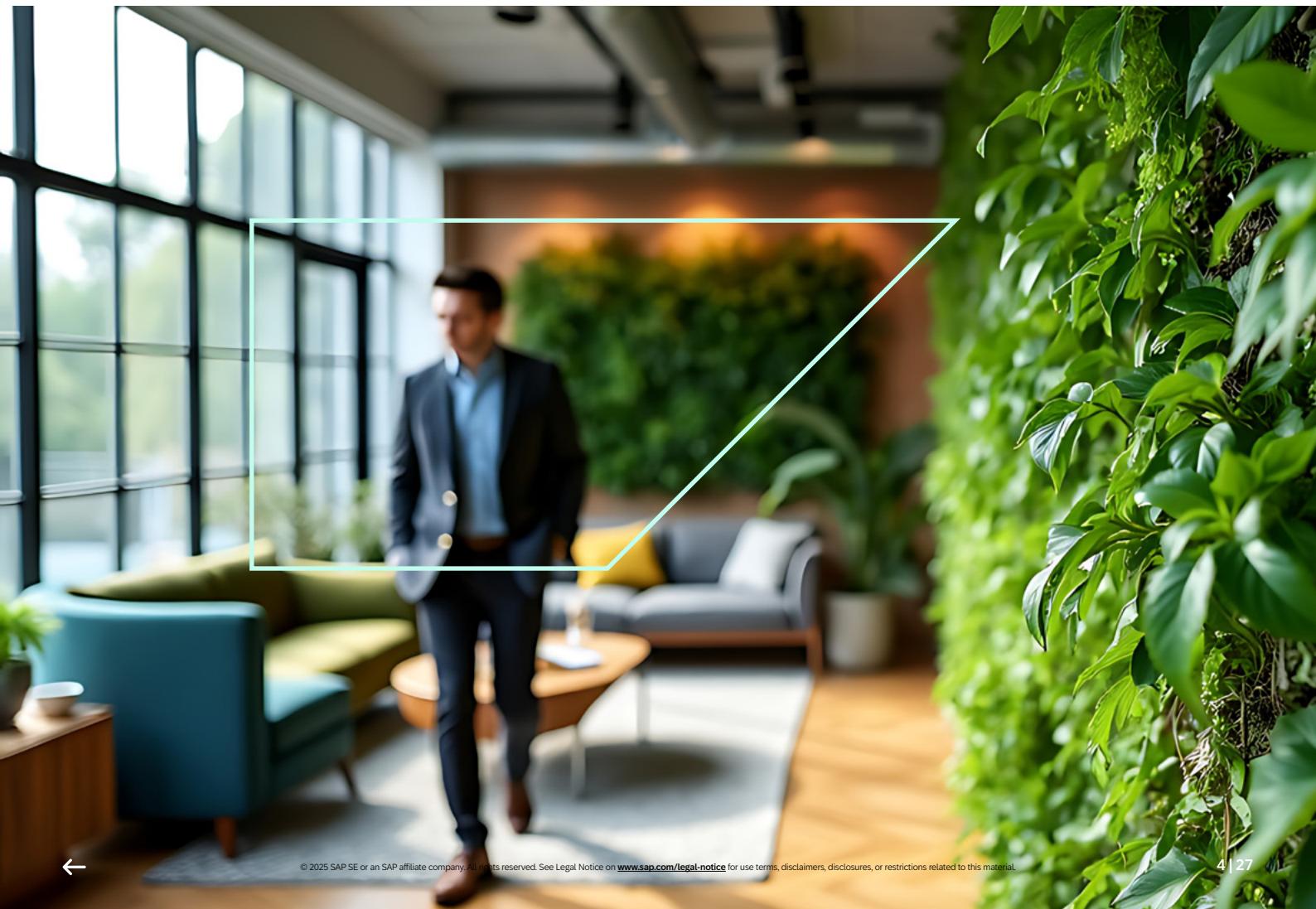
2 The clean core strategy

“Clean core” is a concept and strategy to extend SAP S/4HANA Cloud systems in an upgrade-safe way. SAP S/4HANA Cloud Public Edition does and will continue to have a clean core, as SAP technically enforces the guidelines of the [public cloud extensibility model](#). To achieve or keep a clean core in SAP S/4HANA Cloud Private Edition, certain recommendations and guidelines must be followed (see the paragraph on the [private cloud extensibility model](#)).

Extending SAP solutions to meet specific business needs has always been a significant advantage for SAP customers. Therefore, keeping a clean core, customers experience better maintainability, along

with lower total cost of ownership for SAP S/4HANA Cloud while being agile enough to adapt to change when and where required. A clean core is defined as an SAP S/4HANA Cloud solution that is up to date, documented, unmodified, consistent, efficient, and has custom extensions decoupled from SAP source code. The method for maintaining this clean core while adapting SAP solutions to meet specific business needs is called clean core extensibility, which will be discussed in the following paragraphs. Flexibility will continue to benefit customers in the future, provided that extensions are built and utilized in adherence to clean core principles, the topic of the following section.³

3. Find [more information](#) on the ERP clean core strategy.





2.1 Clean core principles: Guidelines and benefits of extensibility

Building extensions in adherence to clean core principles achieves the balance between software flexibility with customer adjustments and system stability and availability in the cloud. The main guideline to adhere to when building an extension is to keep it strictly separate from the underlying SAP applications. Extensions must access SAP business objects only through well-defined, upgrade-stable interfaces. Extensions calling, for instance, unreleased function modules or accessing the underlying database table directly can easily break as soon as the underlying system code is changed and an upgrade is performed. The clean core approach, however, results not only in faster software deployment but also in easier adoption of software changes, since the core starts off clean and is kept that way using nondisruptive, regularly scheduled upgrades.

Clean core is a framework of best practices that – when followed – results in keeping both the SAP S/4HANA Cloud solution and its extensions upgrade stable. Let's focus on how to achieve this.

Maximize upgrade stability by adhering to these recommended clean core guidelines:

- + Ideally, adopt a policy of zero modifications of the core system.
- + Never clone SAP standard functionality.
- + Retire unused extensions.
- + Adhere to general SAP code quality standards and best practices.⁴
- + Keep all extensions well documented.
- + Set up a strong governance to follow the clean core architecture (further described in the [paragraph on the 3-tier extensibility model](#)).
- + Decouple all extensions from the system's core by leveraging public, released APIs.⁵
- + Leverage all extensibility options, that is, have all extensions running on stack, side by side on SAP Business Technology Platform (SAP BTP) or hybrid extensions residing both on stack and side by side, respectively.
- + Create technical debt only as well-informed decisions and only in cases where there is no other possibility. Document these decisions and revise the technical debt yearly against new possibilities.
- + Keep the SAP software versions close to the latest release (current or n-1), feature pack stack (FPS), and support pack stack (SPS).
- + When possible, use only partner solutions that are certified as clean core compliant.⁶ Discuss your clean core strategy with partners and establish respective governance for the partner solutions.

4. For more detailed information see these blog posts on the [Usage of ABAP Test Cockpit](#) and [ABAP Test Cockpit in the Cloud](#) and the [SAP BTP Developer's Guide](#).

5. Find public, released APIs to integrate and extend on [SAP Business Accelerator Hub](#).

6. For more information, see an [overview of implementation and certification of ABAP Cloud add-ons for SAP Partners](#) or this [blog](#). For more information, see "Extend SAP S/4HANA in the cloud and on premise with ABAP based extensions," SAP brochure, April 2024.



RISE WITH SAP

Driving Business Innovation Together

RISE with SAP Methodology

[RISE with SAP Methodology](#) has been launched to drive these guidelines with customers and focus on the clean core principles. Customers using RISE with SAP should leverage this key entitlement to create a clean core project, establish a solution standardization board, and follow governance recommendations including regular reporting and execution of quality gates.

What are the benefits when following these guidelines?

Agility starts by having a solid foundation in the cloud that leverages the previously mentioned best practices and guidelines. Adopting them brings benefits such as being able to quickly react to changing environments that can include, for instance, a change in your company's strategy or processes, changing customer requirements, or sudden environmental or economic challenges. Having access to SAP's innovations – being part of the latest release and, therefore, requiring an up-to-date core system – adds to that agility. It allows customers to stay on top of new technologies and maximally leverage innovative services such as generative AI, low-code and no-code application development, and process automation.

System upgrade projects will be made significantly easier when clean core extensions don't access the to-be-updated core in nonrecommended ways. And when they do, if there simply has been no other option available, customers will know where to check because the upgrade will be documented well. In addition, system upgrades will eventually be both less complex and costly, as there will be fewer outliers or custom extensions to potentially harm the upgrade process adhering to the highest efficiency and security standards. Moreover, the more regular updates are run, the easier they will get, as best practices will be established and well known and proven project and test plans can be carried out.

3 Extensibility of SAP S/4HANA Cloud

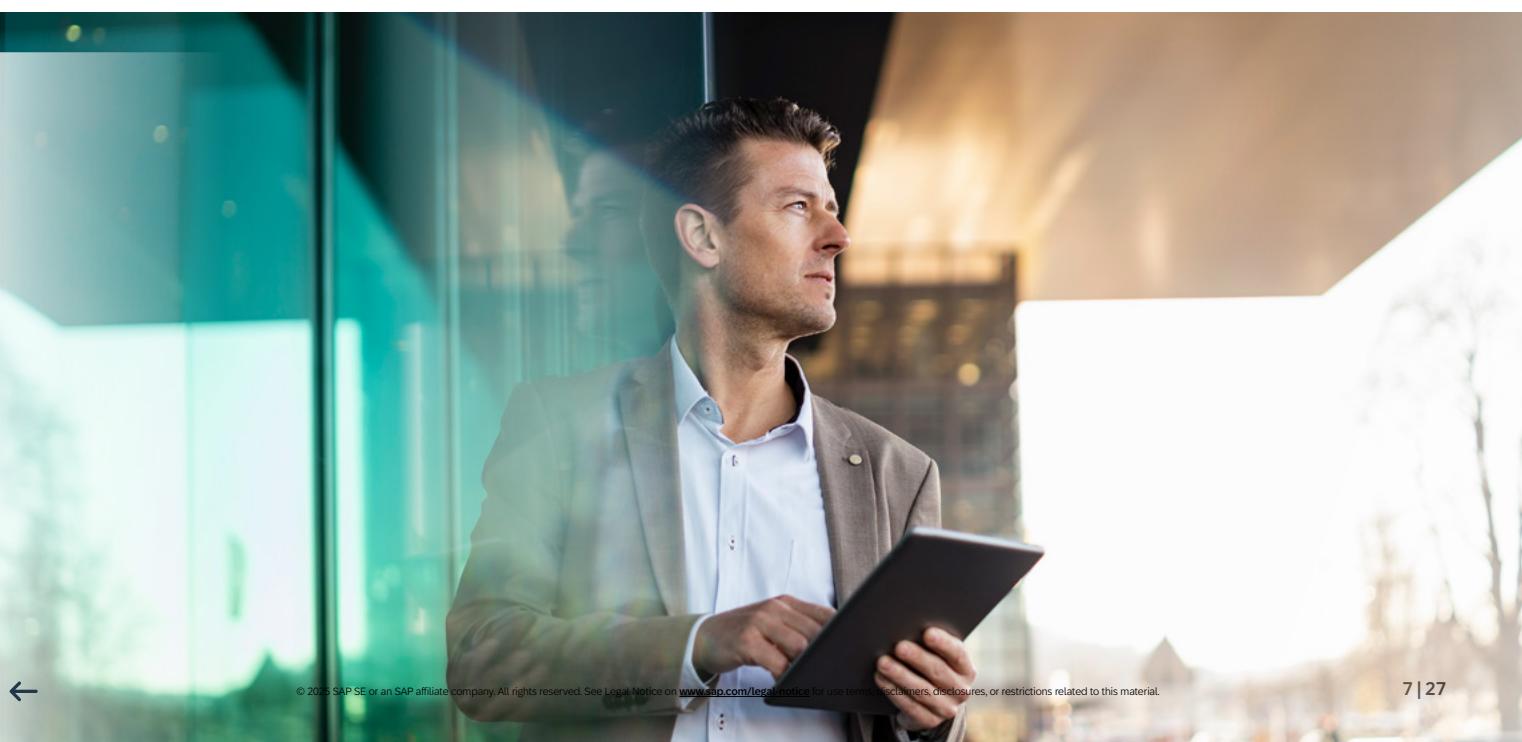
Whichever SAP S/4HANA Cloud solution customers adopt, there must be ways in place that allow for deviation from the standard offering where needed to stay competitive and agile as per the customers' unique business needs. The degree of freedom varies between SAP S/4HANA Cloud Public Edition and SAP S/4HANA Cloud Private Edition, and the core concepts and differences will be explained in this chapter.

SAP offers SAP S/4HANA Cloud Public Edition as a best-in-class public cloud ERP solution paired with a competitive cloud extensibility offering and an established robust partner ecosystem. This public cloud ERP solution features developer extensibility on the code level consuming **Tier 1 extensibility** concepts.

Customers using SAP S/4HANA Cloud Private Edition are supported on their transformation journey with a more flexible extensibility offering that enhances Tier 1 extensibility where it is of utmost importance with so-called Tier 2 and Tier 3 extensibility options. All these options, including the role of SAP BTP in the extensibility context, will be elaborated on further in this chapter.

3.1 The extensibility model

Independent of which cloud ERP solution you choose, be it SAP S/4HANA Cloud Private Edition or SAP S/4HANA Cloud Public Edition, it is standard practice to support extensibility use cases with two different architectural approaches: **on-stack extensibility**, for extensions that are tightly coupled with your ERP core, which means they technically run on the same stack, and **side-by-side extensibility**, for extensions that are loosely coupled with your ERP core and run on a separate extensibility platform. In the SAP context, this means that on-stack extensibility is used in case your extension should be tightly coupled with your SAP S/4HANA Cloud solution, while side-by-side extensibility is used when you want to loosely couple your extension with your ERP system and run it on SAP BTP. Often, a combination of on-stack and side-by-side extensibility is needed to fulfill your business requirement. When to use which extensibility pattern will be detailed next.



3.2 Extensibility options: when to use what?

The following guidelines and recommendations help to determine what part of the extension should run on SAP S/4HANA as an on-stack extension and what part of it should run on SAP BTP as a side-by-side extension. The recommendations apply to SAP S/4HANA Cloud Private Edition, SAP S/4HANA Cloud Public Edition, and SAP S/4HANA. Note that this guidance includes

insights to help navigate the decision-making process effectively, but it is not exhaustive. It is important to analyze each extension use case and make an informed decision about which technologies work best for each case. For more information and support, see [SAP BTP Guidance Framework](#) and use [SAP Application Extension Methodology](#).

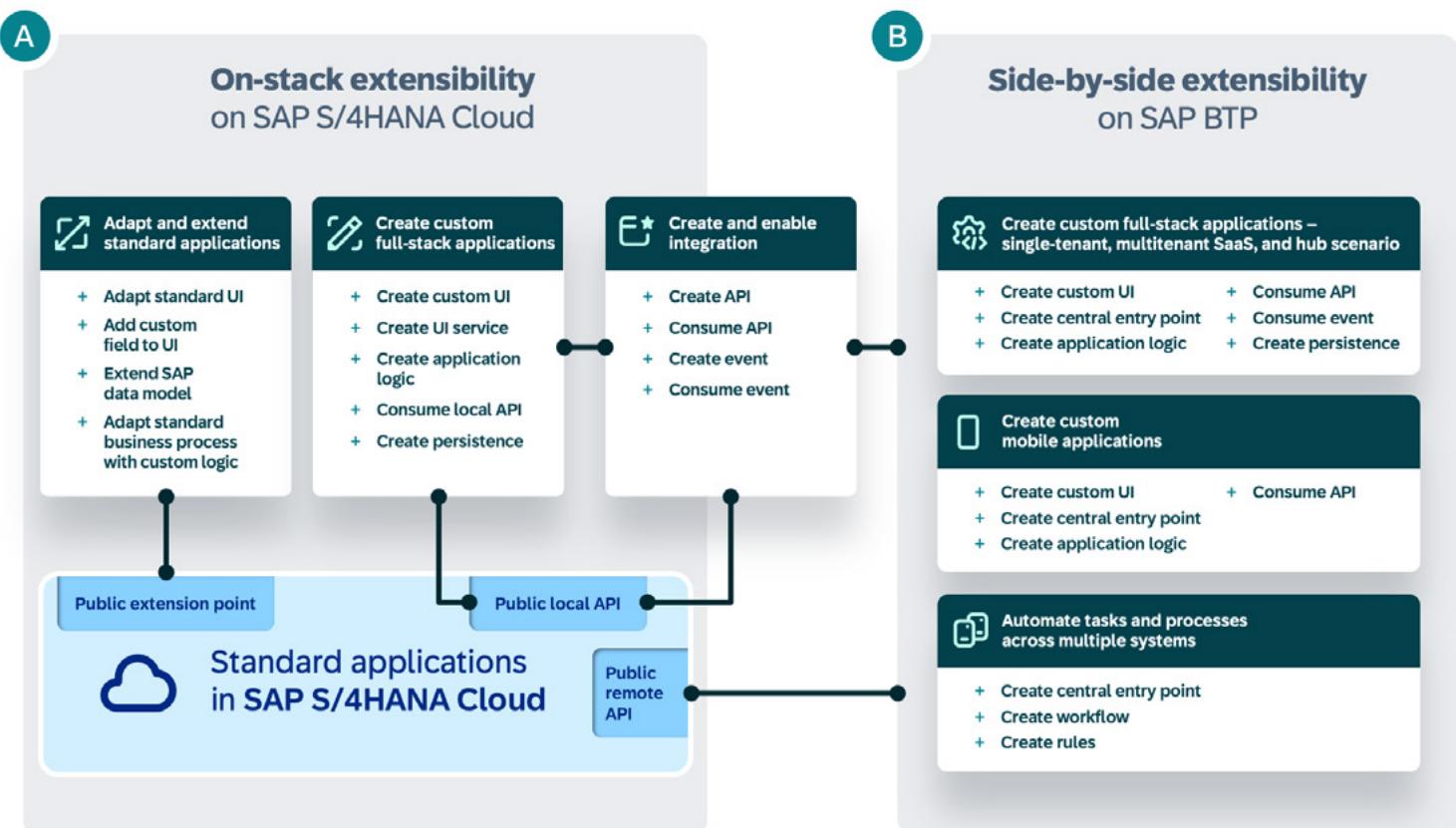


Figure 1: Extension use case patterns for SAP S/4HANA Cloud

Figure 1 depicts common extension use case patterns for SAP S/4HANA Cloud – both for **on-stack** (to the left) and **side-by-side** (on the right) extensions. For more information, see the [extension architecture guide on extension use case patterns](#).

A On-stack extensibility on SAP S/4HANA Cloud

On-stack extensions usually need to be tightly coupled, which means they need proximity to data, transactions, or applications in SAP S/4HANA Cloud.

► Common on-stack use case patterns are as follows:

- ✓ Adapt and extend standard applications:** With this extensibility pattern, you typically perform the following extension tasks:
 - + Adapt a standard user interface (UI)
 - + Add a custom field to a standard UI
 - + Extend the SAP data model
 - + Adapt a standard business process with custom logic within a logical unit of work by implementing the extension point of an SAP application.

- ✓ Create custom, full-stack applications:** This extensibility pattern enables you to develop new applications that require proximity to SAP business objects and services according to the previously mentioned characteristics. Full-stack applications consist of a back- and front-end implementation and, therefore, span the full stack.

- ✓ Create and enable integration:** This extensibility pattern uses public local APIs from SAP or events to build a custom remote API or event, respectively, that can be used for a side-by-side extension on SAP BTP. For more information on public local APIs or events, see this page on [developer extensibility](#).



The following are typical characteristics of an on-stack extension:

- + Transactional consistency is required when data from SAP standard tables and fields and data from custom tables and fields are changed jointly
- + Custom integrations, for instance, extensions of standard objects such as APIs or services needed to enable side-by-side extensions
- + Reading high data volumes, which requires complex Structured Query Language (SQL) queries and joins
- + Frequent access and changes to data from SAP standard tables and fields, resulting in many roundtrips

On-stack extensions on SAP S/4HANA Cloud will be implemented with the ABAP Cloud development model.⁷ The benefits are described in the following paragraph.

7. For more information, see “[Extend SAP S/4HANA in the cloud and on-premise with ABAP based extensions](#),” SAP brochure, April 2024.



The role and benefits of ABAP Cloud in the extension strategy

ABAP Cloud⁸ is the development model for developing cloud-ready business applications, services, and extensions. It is available both on stack in SAP S/4HANA Cloud and also side by side on SAP BTP (see further benefits of creating extensions with ABAP Cloud in the context of SAP BTP in the following paragraph). Some of its benefits are listed below.

ABAP Cloud provides one common development model both on stack and side by side. It's available on SAP S/4HANA Cloud Public Edition, SAP S/4HANA Cloud Private Edition, on premise, and on SAP BTP. It is used to develop customer and partner extensions as well as SaaS applications on SAP BTP. It is the main development model used by SAP itself to drive the development of SAP S/4HANA Cloud. It allows leveraging existing know-how of ABAP development tools for cloud-ready and clean extensibility.

ABAP Cloud also helps ensure cloud readiness and clean core compliance by default. The new language version introduced with ABAP Cloud allows the enforcement of cloud and clean core rules by compiler and syntax checks. Using ABAP Cloud, these rules cannot be evaded.

ABAP Cloud provides a model-driven architecture for online transactional processing (OLTP), online analytical processing (OLAP), and the development of new integrations and APIs by the ABAP RESTful application programming model. Moreover, strong transactional consistency is enforced through the controlled logical unit of work.

ABAP Cloud comes with a sound set of technical and business services (logging, change documents, number ranges, jobs, factory calendar, currency conversion, units of measure, and many more). These services can be used without any additional license and can reduce both total cost of development and ownership for complex business applications. At the same time, they drive a high level of standardization and supportability of custom extensions. Last but not least, default configurations for these services (calendars, countries, languages, currencies, units, and so on) are delivered automatically and are ready to use.

ABAP Cloud contains what you need to help ensure code quality, code documentation, and stable life-cycle management of your custom extensions. It is integrated into the development environment for ABAP Cloud: ABAP development tools.

ABAP Cloud will benefit from the [Joule copilot](#) with ABAP development tools to increase developer efficiency. At the same time, an AI software development kit for ABAP will offer a consumable SAP standard library to conveniently use a generative AI hub on SAP BTP to leverage AI functionality provided there in custom ABAP extensions.

As mentioned, ABAP Cloud is also available on SAP BTP to build side-by-side extensions, which is explained next.

8. For more information on ABAP Cloud, see the [documentation](#) on the SAP Help Portal site.



B Side-by-side extensibility on SAP BTP

[SAP BTP](#) is an integrated offering comprising application development and automation, integration, data and analytics, and AI services. The platform offers users the ability to turn data into business value, compose and automate business processes, and build and – most important – extend applications in SAP S/4HANA quickly. It provides integration capabilities to help ensure business processes are connected across SAP and third-party solutions. SAP BTP is a platform for developing side-by-side extensions that are loosely coupled with the core of SAP S/4HANA. Custom applications or extensions that run on SAP BTP can interact with SAP S/4HANA using APIs and events while not modifying the core.⁹

9. For more information about available SAP standard APIs and events, see [SAP Business Accelerator Hub](#).

▷ Common use case patterns for side-by-side extensions are as follows:

✓ Create custom, full-stack applications:

- + Full-stack, single-tenant applications – This extensibility pattern targets customers and implementation partners, enabling them to develop full-stack apps on SAP BTP.
- + Full-stack, multitenant SaaS applications for partners that are independent software vendors (ISVs) – This extensibility pattern targets the ISV partner ecosystem, enabling them to develop full-stack SaaS applications on SAP BTP and to distribute the software to their customers. For an implementation example of this use case pattern, see the [mission on the SAP Discovery Center site](#).
- + Hub scenario of integrating with several ERP systems and cloud services – This extensibility pattern provides a central hub on SAP BTP to collect and distribute data from various systems. For an implementation example of this use case pattern, see the [samples on github.com](#).

✓ Create custom mobile apps:

This extensibility pattern provides mobile development capabilities. For an implementation example of this use case pattern, see the [mission on SAP Discovery Center](#).

✓ Automate tasks and processes across multiple systems:

This extensibility pattern provides low-code/no-code capabilities to automate processes. For an implementation example of this use case pattern, see the [mission on SAP Discovery Center](#).



The following are typical characteristics and examples for side-by-side extensions:

- + Stand-alone applications or new business process steps that can be decoupled from custom tables and fields that isn't changed together with
- + data from SAP standard tables and fields, therefore, no transaction consistency is required
- + Demand to use a broad spectrum of different development preferences, skill sets, and languages such as Java, JavaScript, or ABAP and freely available open-source libraries
- + Applications for external users without a user account for SAP S/4HANA and who will not have direct access to an SAP S/4HANA solution
- + Benefits of independent infrastructure, scalability operations, and lifecycle with frequent shipments and changes

The services and technologies offered on SAP BTP to implement the previously mentioned extension patterns are described in more detail in the following chapter. More detailed information can be found in both the [extension architecture guide](#) and the [SAP BTP Developer's Guide](#).

The following paragraph summarizes the role and benefits of SAP BTP when it comes to extension developments.



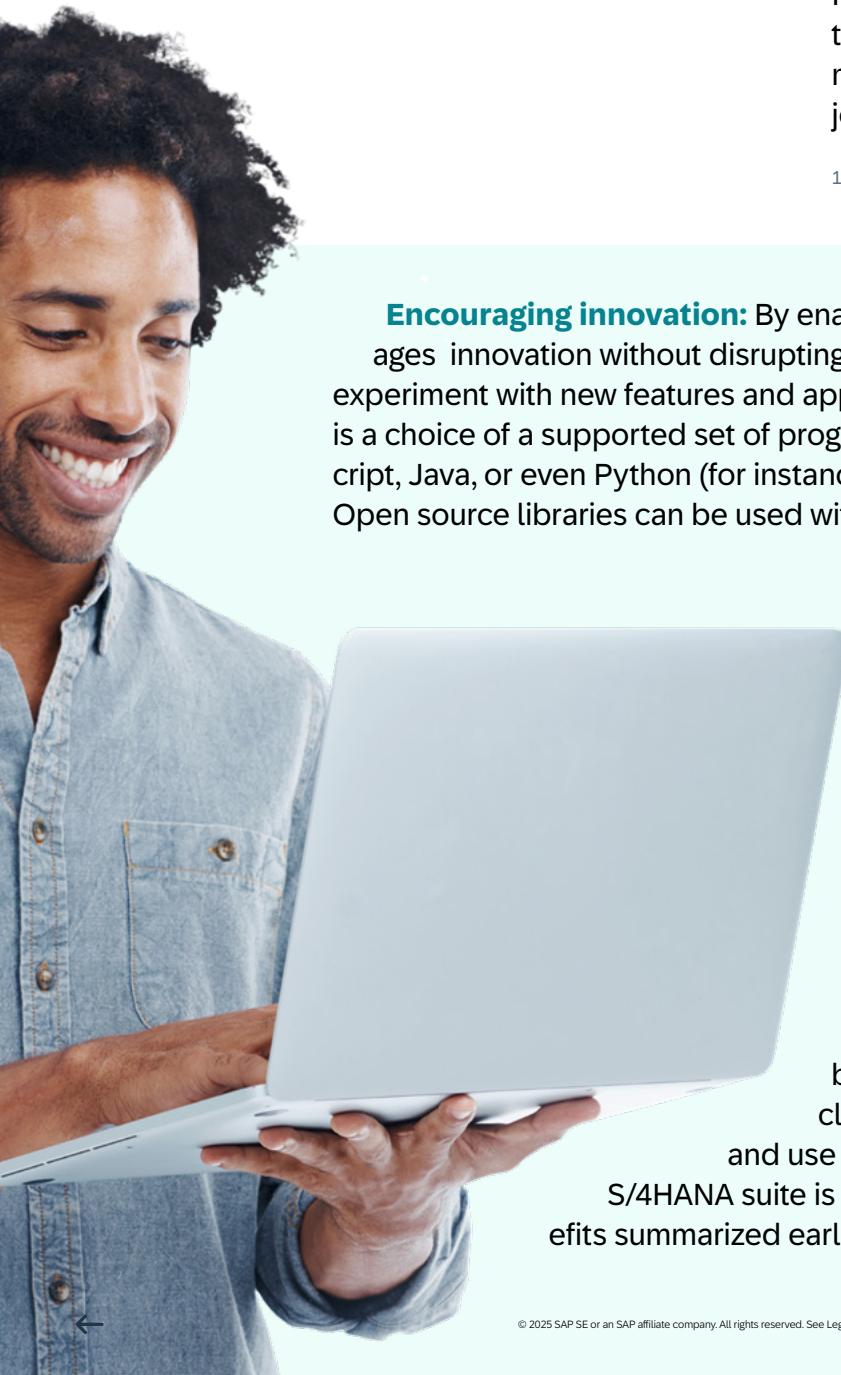


The role and benefits of SAP BTP in the extension strategy

SAP BTP supports two programming models: [SAP Cloud Application Programming Model \(CAP\)](#) to develop in JavaScript and TypeScript or Java, and [ABAP Cloud](#) (described in the [previous section](#)). Both development models are based on common concepts, and they are both centered around core data services (CDS) – a modeling language for domain models and service definitions. Both support creating services that work together especially well in conjunction with UI frameworks such as SAPUI5 and SAP Fiori user experience elements to create enterprise-grade applications in a short amount of time.

CAP services are developed and run on SAP BTP, Cloud Foundry runtime and SAP BTP, Kyma runtime. As mentioned, CAP allows implementation in JavaScript (Node.js) and Java; it offers libraries to implement and consume services, as well as generic provider implementations serving many requests automatically; and it provides the option of using open source libraries to create these services.¹⁰ Services and applications for ABAP Cloud are developed and run in SAP BTP, ABAP environment. Here, ingredients for the development of applications, services, and integrations are included and part of the SAP standard: SAP HANA Cloud, model-driven development using the ABAP RESTful application programming model, technical services, and libraries such as jobs, logs, printing, and XLSX processing.

10. Find [more details](#) on development on SAP BTP.



Encouraging innovation: By enabling side-by-side extensibility, SAP BTP encourages innovation without disrupting the core SAP S/4HANA suite. Developers can experiment with new features and applications on SAP BTP. Creating extensions, there is a choice of a supported set of programming languages such as ABAP Cloud, JavaScript, Java, or even Python (for instance, for AI use cases). Open source libraries can be used within the extensions as well.

Reducing the total cost of ownership:

By using SAP BTP for side-by-side extensions, the total cost of ownership can potentially be reduced. This is because extensions developed on SAP BTP are clean core compliant and loosely coupled from SAP S/4HANA. This will not interfere with upgrades, thereby reducing the effort required for testing and maintenance.

Supporting the transition to a clean core:

By providing a platform for side-by-side extensibility, SAP BTP supports the transition toward a clean core for the previously mentioned scenarios and use cases. While extending side-by-side, the SAP S/4HANA suite is kept unchanged, resulting in the clean core benefits summarized earlier.



The best of both worlds: on-stack and side-by-side extensions

Both on-stack and side-by-side extension options complement each other by providing a comprehensive ecosystem for building, extending, and running enterprise applications. SAP BTP serves as an underlying platform for cloud services and integration for extensions based on ABAP Cloud, CAP, and low-code/no-code technologies coming with respective integrated development environments. In some scenarios, pieces from all of these worlds might be needed. The following are a few examples.

Hybrid deployment capabilities:

Extensibility for SAP S/4HANA often involves a hybrid deployment model, where some components remain on-premise while others are hosted in the cloud. SAP BTP supports this hybrid approach, allowing organizations to deploy extensions in a manner that best suits their needs. Both CAP and ABAP Cloud facilitate cloud-based development and extension while integrating with on-premise SAP S/4HANA software, helping to ensure a cohesive and integrated landscape. As described previously, ABAP Cloud on top gives the option of developer extensibility for tightly coupled extension scenarios.

Hybrid development practices:

SAP BTP promotes modern development practices such as microservices architecture and DevOps methodologies. Both CAP and ABAP Cloud align with these practices by offering cloud-native development capabilities for Java, JavaScript, and ABAP developers. On the ABAP side, this synergy allows developers to adopt modern development paradigms while leveraging their existing ABAP skills, thereby accelerating the extension process for SAP S/4HANA. Fusion team development (as explained in the following chapter), might require developers of different skills to jointly develop an enterprise application. The service built for this might benefit from a tight coupling to the core of SAP S/4HANA, therefore, built using Tier 1 extension options on stack (that is, the ABAP development environment). Another developer might create a mobile UI based on that service with SAP Fiori tools in SAP Build Code – maybe even making use of the available generative AI capabilities enhancing productivity. The resulting full-stack application will be hosted partly on stack and partly side by side.

Integration and connectivity:

SAP BTP provides robust integration capabilities for connecting SAP S/4HANA with other SAP and third-party systems. ABAP Cloud integrates with SAP BTP's integration services, enabling developers to orchestrate complex integration scenarios involving extensions and other cloud services based on ABAP. This integration helps ensure smooth data exchange and interoperability across the landscape.

To summarize, both on-stack and side-by-side technologies (and those available in both environments) complement each other effectively in the transformation journey to SAP S/4HANA. Together, they empower organizations to extend and customize SAP S/4HANA efficiently, adopting modern ways of working and accommodating preferences, thereby meeting evolving business requirements and driving digital innovation.



The role and benefits of SAP BTP in the context of SAP S/4HANA Cloud

SAP BTP plays an essential role in the strategy for SAP cloud ERP solutions. SAP S/4HANA Cloud and SAP BTP are inseparable, as SAP is developing major parts of its innovations as modular apps on SAP BTP, such as SAP Green Ledger, SAP Digital Manufacturing, SAP Advanced Financial Closing, sustainability management, and next-generation industry solutions. SAP is following this strategy to strengthen its clean core strategy:

- + The core holds the essential business processes that companies need, regardless of the industry they operate in (core processes).
- + Differentiating processes (edge, industry, innovative, next-generation, and more) are moved as modular apps toward SAP BTP, establishing modular cloud ERP.

Customers and partners should follow the same strategy and establish two-layer IT:

- + **Layer 1:** A stable clean core holding essential core processes that keep the business up and running, automating processes to lower the bottom line
- + **Layer 2:** An innovation layer with fast innovations, flexibility, modularity, and nondisruption to the core – providing processes that differentiate from competitors and give competitive advantage, increasing the top line (examples include sales-order automation, just-in-time production, and customer loyalty)

SAP BTP is an innovation layer next to a stable clean core and provides superior benefits:

- + It is deeply integrated and interwoven with your core ERP and understands and shares the underlying meta and business model.
- + SAP S/4HANA uses services for SAP BTP (for example, SAP HANA service for SAP BTP, SAP Integration Suite, SAP Datasphere solution, SAP AI Core infrastructure, and ABAP Cloud).
- + Deep integration – An extension wizard can start directly from SAP S/4HANA, which allows you to navigate to tools for extension creation on SAP BTP ([see Chapter 4](#)) or to the service center where we show APIs and business application programming interfaces, along with remote function calls, as such.
- + ABAP Cloud and CAP – Hyperscalers don't offer programming models that help create business applications with that much built-in functionality.
- + [SAP Business Accelerator Hub](#) provides a vast amount of built-in, up-to-date content for our applications, such as integrations, connectors, APIs, localization, and so on.
- + As mentioned in more detail in [Chapter 4](#), Joule, SAP's AI copilot, offers support specialized in SAP software-oriented development (both for ABAP Cloud and CAP).

3.3 The public cloud extensibility model

The previous paragraphs described different extension patterns and ways to approach these for both on-stack and side-by-side extensibility. Now, let's inspect how this is realized for SAP S/4HANA Cloud Public Edition.

On-stack key user extensibility

This extensibility technique provides key users (business experts) with the ability to do low-code/no-code extensions directly on stack running on SAP S/4HANA Cloud Public Edition. To develop these extensions, key users can either use dedicated SAP Fiori apps or make changes directly inside existing SAP business applications.

Examples include:

- + Changing labels and positions in existing SAP business applications or adding custom fields to them
- + Implementing business logic extensions at well-defined locations inside the SAP business objects
- + Creating custom business objects or analytical queries specific to the business department

On-stack developer extensibility

Developer extensibility offers professional, comprehensive development of extensions, applications, and integrations using the ABAP Cloud development model. These extensions run on the stack for SAP S/4HANA and can leverage

local APIs from SAP, such as CDS views, interfaces, business objects, and the like. Developers use ABAP development tools for Eclipse in their development projects and can make use of full-fledged development capabilities including profiling, debugging, and code checks. By using only released public APIs, extensions developed with ABAP Cloud are upgrade stable and have a clean core by default. The list of released APIs can be found in ABAP development tools but also on SAP Business Accelerator Hub and in the [cloudification repository on GitHub](#). Further information is provided on [SAP Community](#).

Side-by-side extensibility on SAP BTP

Side-by-side extensions are developed on SAP BTP and have a clean core by default. Communication with SAP S/4HANA Cloud solutions is possible through released remote APIs (such as OData APIs) and events, which can be found on SAP Business Accelerator Hub. The portfolio of services, tools, frameworks, and runtimes for SAP BTP offer a lot of flexibility and target different development and technology preferences and personas. Tools are available that address both key users (business experts) and developers. As mentioned previously, side-by-side extensibility on SAP BTP can be based on two development models – either [ABAP Cloud](#) or [CAP](#).



3.4 The private cloud extensibility model

The public cloud extensibility model described above is reused in SAP S/4HANA Cloud Private Edition. It is cloud enabled and has a clean core by default and is, therefore, the go-to extensibility model for the private cloud including key user, developer, and side-by-side extensibility. In the context of the private cloud, these extensions are called Tier 1 extensions. One of the key principles of Tier 1 extensions is the usage of released APIs. If such an API is missing, a cloud

API enablement technique exists in the private cloud, referred to as Tier 2. Furthermore, the private cloud allows for brownfield implementations and conversion from SAP ERP Central Component to SAP S/4HANA Cloud Private Edition and, therefore, allows the takeover of classic extensions for ABAP. These will be collected in Tier 3. The interplay of these extensibility tiers is called the 3-tier extensibility model, shown in Figure 2 below and explained in the next subsections.

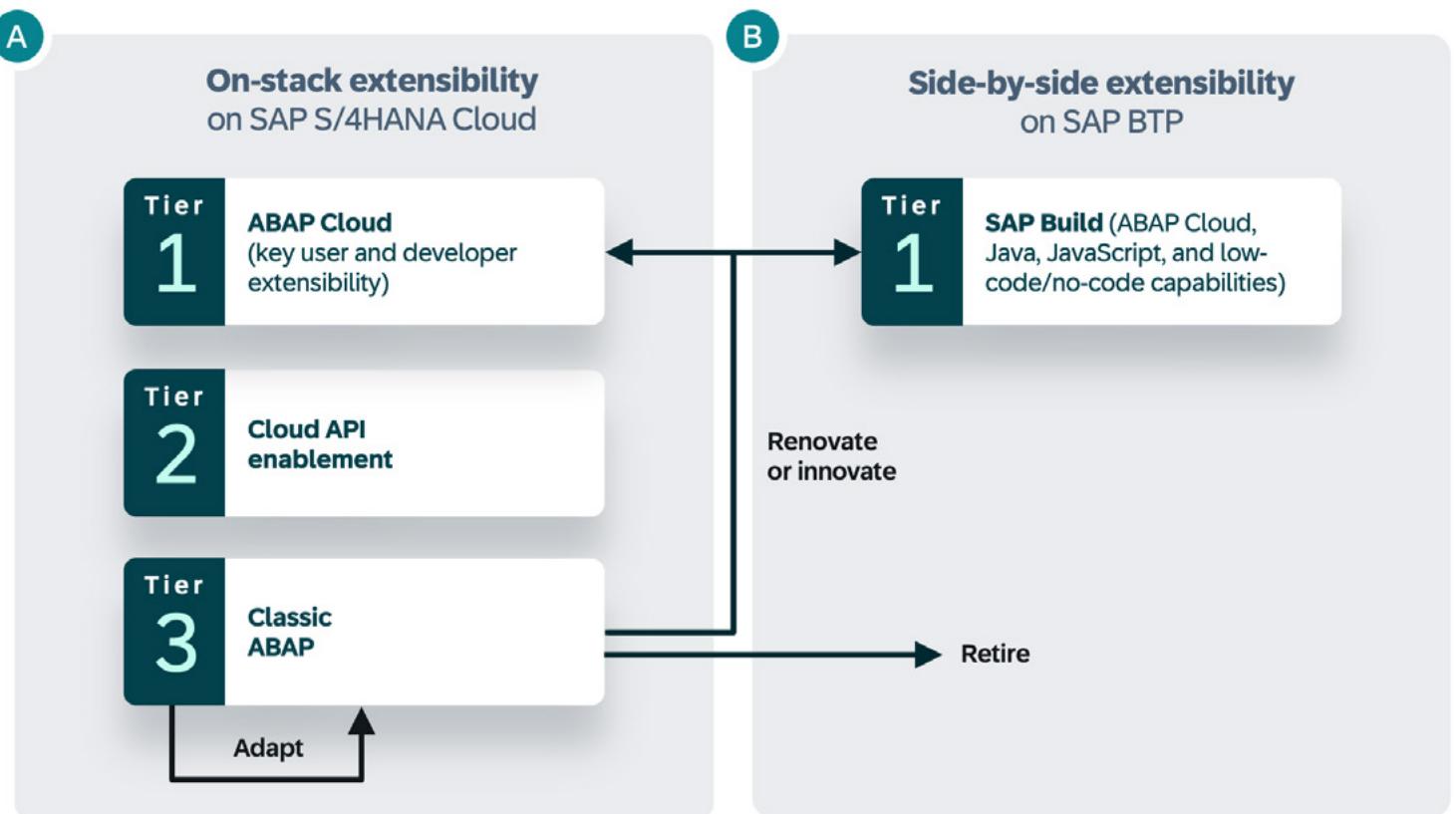


Figure 2: The 3-tier extensibility model for SAP S/4HANA

3.4.1. Tier 1

Tier 1 extensions are developed using ABAP Cloud for on-stack extensions and all capabilities offered by SAP BTP (see [Chapter 4](#)). Thereby, they are strictly decoupled from the SAP code through released APIs, events, and extensions points. This is the go-to extensibility tier for new extensions.

3.4.2. Tier 2

There are cases where the required public APIs for Tier 1 extensions are missing. To unblock this situation, Tier 2 provides an API enablement mechanism to make use of classic APIs instead. This can be the case either because the public APIs come with a higher release, or it can also occur in areas where SAP does not actively invest in new APIs (for example, classic HCM). In these cases, SAP nominates existing classic APIs.



3.4.3. Tier 3

Tier 3 contains classic extensions for ABAP. Hence, existing classic code in ABAP brought into SAP S/4HANA Cloud Private Edition, for instance, from the conversion from SAP ERP Central Component, will reside in this tier. There is no “magic button” that allows the transformation of all existing extensions for ABAP to Tier 1 and Tier 2. Moreover, there are cases where extensions cannot be developed in Tier 1 due to technical limitations (such as extensions to SAP business applications based on the SAP GUI interface). Hence, we recommend starting the journey to reach a clean core by following three steps:

- 1** Retire all unused custom code and remove it from the system. Perform a fit-to-standard analysis¹¹ to check which existing extension can be replaced by SAP standard functionalities.
- 2** If there is a business case to renovate an extension (for example, to make it future ready for the cloud, make use of new technologies or mobile apps, and so on), bring it step by step toward Tier 1 and Tier 2.
- 3** For the rest, ensure minimum quality and upgrade stability by following best practices to adapt your code. To do this, leverage ABAP test cockpit and classic APIs.

For details and more information, please see the [extensibility guide for ABAP](#).

3.4.4. Recommendations for partners

Partners should follow the same recommendations for their partner solutions and add-ons as customers, so the above guidelines also hold for partners. Partner solutions and add-ons should not break the clean core strategy of customers. To this end, a clean core certification for partners has been created. Customers can find certified partner solutions for a clean core in the [SAP Certified Solutions Directory](#) mobile app.

¹¹ For more information, see this [premium engagement session](#) on intelligent custom code management or this [support page](#) on fit-to-standard workshops.



3.5 Launching a clean-core certification program for partner applications

[SAP Integration and Certification Center \(SAP ICC\)](#), part of the Partner Innovation Lifecycle Services group, announces the launch of a certification program for partner applications following the clean core criteria, with which it is possible to prove clean core compatibility to customers.

With the clean core approach, SAP wants to make sure that partner applications follow the strategy in both the development and delivery of partner solutions to SAP customers. In the optimal case, this is enabled by using SAP recommended extension options only, as described in the [extensibility model for SAP S/4HANA Cloud](#). The following partner apps that previously adhered to the 3-tier model outlined in this document can earn the clean core designation:

Add-ons for ABAP Cloud (Tier 1 or Tier 2) built for SAP S/4HANA Cloud Private Edition¹²
Side-by-side solutions running on SAP BTP (Tier 1) and integrating with SAP S/4HANA Cloud Private Edition. Technical feasibility will be checked and confirmed through [Partner application on SAP BTP Certification](#). Further technical details regarding the certification program can be found in this [presentation](#).

When starting with SAP S/4HANA Cloud in a greenfield approach, that is, without an existing structure or processes, we recommend you follow the previously mentioned guidelines and recommendations. That means using Tier 1 extensibility where possible, Tier 2 extensibility when no publicly released SAP APIs are available, and Tier 3 extensibility for your existing classic extensions for ABAP or when using Tier 1 and Tier 2 is not possible. For existing classic extensions for ABAP, follow the described steps for Tier 3 to reach a clean core step by step.

12. The certification scenario is generally available, and if you want to receive more details, please contact us at icc@sap.com.

3.6 APIs, events, and extension points

Publicly released APIs, events, and extension points are the foundation for clean core extensibility both on stack and side by side, since they allow decoupling from SAP code. They can be found on [SAP Business Accelerator Hub](#). Customers and partners can request new APIs, events, and extension points through Customer Influence program channels as shown below.

SAP S/4HANA Cloud Public Edition:

- ◀ [SAP S/4HANA Cloud and SAP S/4HANA for Integration](#)
- ◀ [SAP S/4HANA Cloud and SAP S/4HANA for Key User and Developer Extensibility](#)

SAP S/4HANA Cloud Private Edition:

- ◀ [SAP S/4HANA Cloud Private Edition for Extensibility and Integration \(APIs\)](#)

On the [SAP Road Map Explorer](#) Web site, it is possible to choose a certain product (for instance, SAP S/4HANA Cloud Public Edition) and select the focus topic filter **Integration** to learn about upcoming APIs.

Now that the strategy and the reasoning behind the clean core has been laid out, the following chapter describes how to develop extensions achieving and keeping a clean core, that is, which services support extension development according to the clean core strategy and how best to leverage the benefits of the SAP Cloud Application Programming Model and the ABAP RESTful application programming model for both on-stack and side-by-side extensions.



4 Extend SAP S/4HANA with SAP Build

The go-to place following SAP-recommended best practices to create extensions optimized for SAP software development is [SAP Build](#). SAP Build includes powerful low-code, pro-code (ABAP, Java, JavaScript), and generative AI capabilities, and enhanced interoperability with SAP S/4HANA to empower developers and business experts to create extensions for SAP software with greater efficiency. It allows customers to accelerate ERP modernization, foster innovation, and automate processes – all within a single comprehensive suite of solutions.

Streamline extension of SAP S/4HANA

Through application development and process automation with SAP Build, the previously described programming models CAP and ABAP Cloud can be leveraged, being aligned with clean core principles to create scalable, secure, and stable extensions.

Accelerate extension of SAP S/4HANA with AI capabilities enabled by Joule

New levels of ERP efficiency can be unlocked with the latest AI capabilities enabled by Joule in SAP Build. Joule is SAP's AI copilot offering conversational interactions in the respective business context. Uniquely trained on SAP data and processes, it helps developers write code and design workflows for SAP S/4HANA across ABAP, Java, JavaScript, and low-code tools. As an integral part of SAP Build, Joule is the copilot for SAP development and makes developers more efficient with their development tasks for SAP S/4HANA.

Joule generates high-quality code and code explanations aligned with SAP's programming models, thereby reducing development time for both new and experienced developers and accelerating migration to cloud ERP with a clean core.

AI capabilities enabled by Joule can also be used to extend ERP through low-code application development. Joule assists with process automation, expediting the creation of workflows for SAP S/4HANA and guiding workflow approvers with automated recommendations.

Fusion team development

SAP Build offers diverse, multiskilled teams with professional developers and business experts what they need to create enterprise extensions fostering collaboration, asset sharing, and seamless technology bridges with built-in governance and lifecycle management (see Figure 3).

The [lobby for SAP Build](#) is the central place and the starting point to begin an extension project with SAP Build – be it for side-by-side extension development on SAP BTP or for on-stack extension development. Depending on the use case

of the extension, the respective development environment can be launched as, for instance, in a complete application development pro-code scenario, ABAP development tools for an on-stack-extension or SAP Build Code development tools, or ABAP development tools for a side-by-side extension. The lobby provides many convenient features, such as monitoring capabilities and access to prebuilt content to supercharge developer productivity.

SAP Build: Generative AI-powered app development and automation on SAP BTP

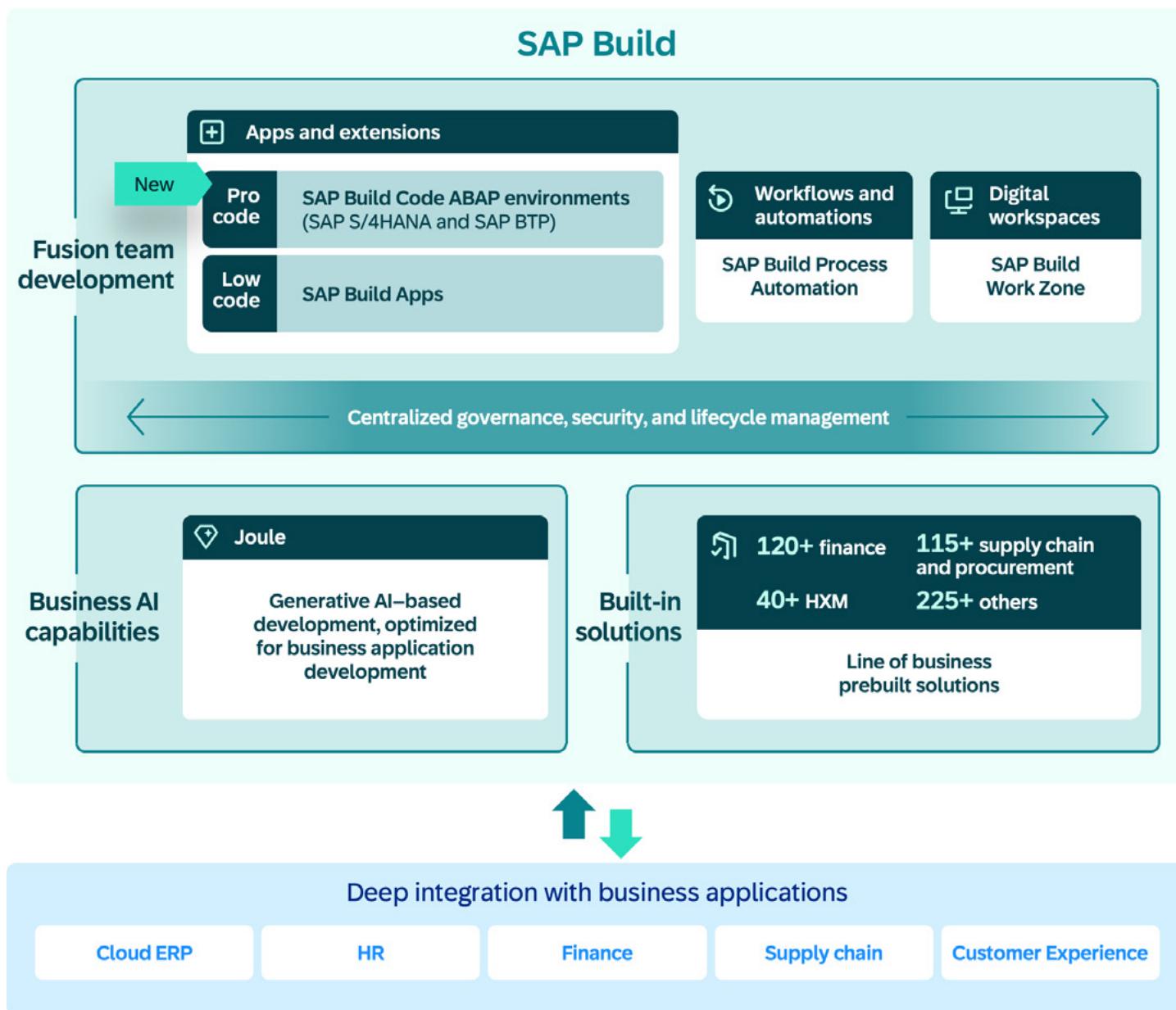


Figure 3: SAP Build – what it comprises, how it can be used, and which solutions can benefit from it

5 Getting started

Your path to a clean core – as a step-by-step guide on how to get and stay clean – is similar for both greenfield and brownfield implementations. Whereas for greenfield the main task is to design a system utilizing as much standard functionality as possible and implement governance to **keep** this system clean (all steps except step 3 on the following page), a brownfield implementation – in addition – entails a transformation journey that strives to **make** existing custom code clean (all steps including step 3 on the next page). SAP recommends anchoring the clean core principles in an organization's IT strategy and follow clean core guidelines as mandatory for new extensions

– whether in a greenfield or brownfield approach. [RISE with SAP Methodology](#) embeds these into the customer journey (greenfield or brownfield) through relevant tasks across project phases, in the form of a standardized framework, governance, guidance, and use of a tool chain. Quality gates, as well as reports with recommendations and actions, help you stay on track during your transformation journey, adhering to clean core principles. Whether through SAP-led or partner-led engagements, you can get a clear path with specific implementation stages and success parameters to be able to take advantage of speed, agility, and the benefits of the cloud.



The six-step path to a clean core is as follows:

- 1 Start with the clean core concept and definition:** This document is a perfect place to start, as it provides explanations and recommendations, with links to learn more about and understand the concept. Please refer to this [document](#) and to the SAP Learning Journey “[Applying the RISE with SAP Methodology](#)” to help understand how you would apply these concepts in your transformation journey in a tangible way.

- 2 Apply the concept and create your specific clean core strategy:** The clean core concept needs to be applied to the specific requirements of each customer and should be reflected in all aspects of your IT strategy. This [presentation](#) provides further details to support this.

- 3 Analyze existing custom code (brownfield only):** As discussed in [Chapter 3](#), existing custom code needs to be [analyzed](#). You should prioritize what to touch first, retire unused objects, and create a list of objects to be redesigned or replatformed. Further tools to help with this are described in [this blog post](#) and this [overview](#).

- 4 Adopt extension guidelines and methodologies:** Conduct a fit-gap analysis resulting in a gap list. Ensure fit to standard, including validating gaps against future product road maps, and design clean extensions following the [SAP Application Extension Methodology](#) and [SAP Integration Solution Advisory Methodology](#) according to the previously described guidelines.¹³

- 5 Build extensions:** Follow the 3-tier model described in [Chapter 3](#) and use the services described in [Chapter 4](#) to develop on stack or side by side, keeping the core clean.

- 6 Establish clean core governance and operations:** As mentioned previously, document your technical debt, establish clean core governance, and embed your undertakings in your business transformation journey.¹⁴ Clean core governance is key to success. Every deviation from clean core principles (or your specific clean core strategy) should go through a governance board that has to approve and document every exception. Clean core baselines such as KPIs and thresholds should also be in place, as should periodic monitoring using technical governance tools, such as mandatory checks from the ABAP test cockpit. The goal is to minimize the number of business modifications and unclean and unused custom code and maximize the share of code that is clean core compliant.

13. Find more information: [Extensibility guide for ABAP](#), [fit-to-standard analysis for SAP Signavio solutions](#), [SAP BTP Guidance Framework](#), [SAP Extensibility Explorer tool](#)

14. [SAP LeanIX solutions](#) and the [SAP Cloud ALM solution](#) can help with this.

6 Summary

Striving for a clean core comes with many benefits. However, it certainly poses several challenges. We continue to try to solve these challenges. We do so by providing a rich extensibility offering for both SAP S/4HANA Cloud Public Edition and SAP S/4HANA Cloud Private Edition, considering each of their different backgrounds and requirements. We certainly take into account cus-

tomer needs during this transformation journey. We understand your requirements during your transformation into the cloud, and our extensibility offering will continue to improve to mirror this in the future. Be sure to stay up to date by checking our packed road maps for future innovations to improve the clean core extensibility model for SAP S/4HANA Cloud even further.



7 Appendix

Below is a list of relevant links that can support you on your journey toward a clean core.

Links	Descriptions
Extension use case patterns	Provides help to determine which part of the extension runs on SAP S/4HANA as an on-stack extension and which part runs on SAP BTP as a side-by-side extension
SAP Application Extension Methodology	Describes a structured, technology-agnostic approach for customers and partners to define an organization-specific extension strategy
SAP BTP Developer's Guide	Helps to implement business applications on SAP BTP
SAP BTP Guidance Framework	Provides a central access point for architects, developers, and administrators to build and run enterprise-grade solutions on SAP BTP
SAP Business Accelerator Hub	Offers what customers need to accelerate integrations, extensions, and innovations (APIs, events, CDS views, prebuilt automation content, and more)
Clean core on SAP.com	Gives an overview of the ERP clean core strategy
RISE with SAP Methodology	Represents a comprehensive framework designed to drive digital transformation and intelligent operations within organizations, leveraging advanced tools and strategies to streamline processes, enhance efficiency, and accelerate innovation
From Classic ABAP to ABAP Cloud and clean core	Gives an overview of implementation and certification for add-ons for ABAP Cloud for SAP partners
Extend SAP S/4HANA in the cloud and on premise with ABAP-based extensions	Provides an overview on the extensibility options for ABAP-based extensions on SAP S/4HANA in the cloud and on premise, along with guidelines for project managers, key users, and ABAP developers
SAP Discovery Center	Lets you explore missions, reference architectures, and services for SAP BTP

Links	Descriptions
Extension architecture guide	Shows how you can use the guide to determine which SAP technology should be applied in a given scenario
SAP BTP Developer's Guide	Shows how you can use the guide to improve the process of implementing a business application on SAP BTP
ABAP Cloud	Details a development model for ABAP to build cloud-ready business apps, services, and extensions
SAP Cloud Application Programming Model	Helps you build cloud-native applications with maximized productivity at minimized costs, based on proven best practices
Tutorials for SAP Build Apps	Provides tutorials for SAP Build Apps
Tutorials for SAP Build Process Automation	Provides tutorials for SAP Build Process Automation
Tutorials for SAP Build Work Zone	Provides tutorials for SAP Build Work Zone
SAP Build Code Demo	Showcases SAP Build Code development tools: AI code generation with Joule copilot, built on SAP BTP, that is optimized for Java and JavaScript development; delivering a turnkey environment for coding, testing, integrations, and application lifecycle management.
	Watch the video to get an overview of SAP Build Code.
SAP BTP, ABAP environment on SAP Community	Informs about a platform-as-a-service offering that enables developers to use their traditional on-premise knowledge of ABAP to develop and run applications on SAP BTP
Hitachi High-Tech customer story	Shows how Hitachi High-Tech mastered side-by-side development techniques to innovate in a more agile and future-ready way
SAP Innovation Awards 2024 – Accenture	Gives information about RISE with SAP and the clean core journey of Accenture