Red Hat ROSA: A New Home for Custom Code

White paper

One of the biggest challenges faced by organisations migrating from SAP® ERP Central Component (ECC) to SAP S/4HANA is what to do with custom code. While some can be discarded, many other extensions will need to be kept.

This white paper outlines how custom code migration can be approached through SAP's recommendation to 'keep the core clean', running retained custom code on Red Hat® OpenShift® Service on AWS (ROSA). This architecture not only enables code migration while connecting to the SAP S/4HANA backend but allows organisations to integrate this within a broader cloud-native digitalisation architecture.

SPONSORED BY

CIO

Red Hat | aws

White paper

In 2015, SAP announced the most far-reaching overhaul of its software platform since the company established itself with R/3 in the 1980s. The resulting SAP S/4HANA platform required customers to move from the SAP ERP Central Component (ECC) platform, which has formed the backbone of their ERP for decades to this new platform offering a path to business digitalisation made possible by cloud technologies.

The 'S' in S/4HANA was a reference to the world 'simple', but larger organisations were under no illusions – despite the benefits of moving to a new cloud-native platform, SAP S/4HANA migration would be technically complex and take years to complete. After revising its deadline more than once, the date by which migration must be completed is now 2027, beyond which support for SAP ECC and third-party databases will become expensive, limited and eventually cease.

Generalising about SAP S/4HANA migration is difficult; each organisation faces challenges unique to its operations. But every organisation must choose a platform to migrate to and assess the tools they will need to complete this journey. They must also consider whether they possess the skills necessary to run a migration project over such a long timescale. Most organisations engage a service provider to help with migration but even with assistance it remains a daunting undertaking.

At the start, every migration project must consider several important issues, including:

- Formulating a migration plan that minimises cost uncertainty
- Minimising business disruption by shortening the upgrade timescale as much as possible
- Assessing SAP S/4HANA migration within the context of cloud-native application development
- Considering the issue of long-term infrastructure costs and skills shortages

The challenge of custom code

A critical requirement is what to do with the sometimes large volumes of custom SAP Advanced Business Application Programming (ABAP) code and extensions accumulated over time in the SAP backend. Custom code

White paper

and extensions are absolutely central to how SAP works. Written in languages including Java and Python in addition to ABAP, they add numerous tweaks and bespoke features to SAP applications, including custom analytics, integration with external systems and extensions to database forms.

SHUTTERSTOCK

What to do with custom code can quickly become a major migration barrier. Organisations that have been using SAP for any length of time will have created a lot of custom code, some of which will not work in SAP S/4HANA as it did in SAP ECC. The fact that the custom code will have been developed in multiple languages on different development platforms only adds to this complication.

With SAP ECC, the presence of custom code made upgrading to new versions more complex. Through SAP S/4HANA, the company intended to simplify this core to ease future upgrades; however, the price for this is that customers must rationalise their custom code, discarding code they no longer use, or which will no longer work on SAP S/4HANA.

Greenfield or hybrid?

The first task is to assess the state of custom code by taking an inventory, working out what should be kept, what should be discarded and what must be rewritten from scratch. Smaller organisations with less SAP investment might resolve to abandon custom and legacy code altogether and move to SAP S/4HANA as a new greenfield implementation. The second and more demanding option is to adopt a brownfield approach, rewriting almost all custom code before migrating to SAP S/4HANA. The final hybrid option is to migrate custom code selectively.

For practical reasons, the latter option is usually the one chosen by most organisations. The whole point of SAP S/4HANA is to take advantage of its new features without being over-encumbered by a development cycle oriented towards the past.

Equally, some customisation will still be needed in the new environment to integrate legacy systems and established business processes. For these organisations, abandoning custom code entirely is not an option.

Keeping the core 'clean'

At the heart of custom code migration is SAP's concept of 'keeping the core clean', that is, separating or decoupling custom code from the core SAP backend. Doing this not only rationalises the size of the code base going forward but also hugely simplifies future SAP S/4HANA upgrades. The issue of a simplified or 'clean' core is one reason SAP felt it was necessary to achieve a clean break in its move to SAP S/4HANA. As noted above, adding more custom code over time generated unwanted complexity and higher upgrade costs. The solution is to move custom code to a new platform developed and managed independently from the SAP backend.

Finding a new home

The next task is to find a new platform for custom code that meets an organisation's development priorities while preserving their investment in SAP. For some, this will

Selecting an application platform on which to run SAP S/4HANA custom code is a strategic decision with long-term consequences. One option is to use Red Hat® Linux-based OpenShift containerisation platform.

be SAP's own Business Technology Platform (BTP), an approach to SAP S/4HANA migration that suits organisations whose development is overwhelmingly SAP oriented.

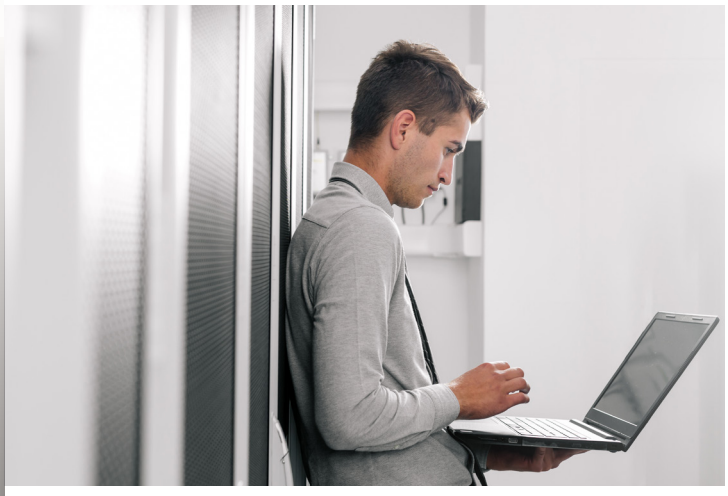
Equally, for many organisations the need to integrate SAP within a wider cloud-native application environment will be more important. These organisations use SAP but not only SAP, building cloud applications using technologies such as Kubernetes containerisation and microservices. The same principle can be applied to custom code. Ideally, this implies that organisations should migrate and run custom code on a container platform that provides the necessary runtimes, software development kits (SDKs) and application programming interfaces (APIs) to allow the custom code to connect back to SAP S/4HANA.

White paper

On-premise vs managed service

Selecting an application platform on which to run SAP S/4HANA custom code is a strategic decision with long-term consequences. One option is to use Red Hat's Linux-based OpenShift containerisation platform. This offers a degree of flexibility for complex migrations which will suit organisations that want to operate SAP S/4HANA in-house as part of a multi-cloud strategy.

The alternative is to opt for a managed service. Deciding which path to take as part of SAP S/4HANA migration involves assessing the pros and cons of each deployment. On the one hand, on-premise development offers a lot of flexibility for larger organisations that have already invested in their own infrastructure and skills. For others, however, the same factors can act to slow their migration path. Skills are often in short supply. The same applies to infrastructure, which forces organisations to make decisions ahead of time about the resources they will need based on assumptions that can quickly become out of date.



SHUTTERSTOCK

Red Hat OpenShift Service on AWS (ROSA)

One of Red Hat's managed solutions is the Red Hat OpenShift Service on AWS (ROSA). Developed in collaboration with Amazon Web Services, the principle behind ROSA is to provide the same set of Kubernetes container-based features offered by an on-premise Red Hat OpenShift in the form of a fully managed service that runs on the Amazon Web Services public cloud (AWS) platform. The architecture this entails is simple: custom code runs inside containers inside ROSA, connecting to the SAP backend (on premise, any cloud or hybrid clouds) and additional SAP services such as BTP platform as a service as required. Meanwhile, Red Hat integration can be used to integrate extensions with the SAP backend through a developer-friendly architecture based on APIs.

ROSA's design not only provides customers with a cloud-native platform in which to run their code, it allows them to achieve this without having to manage the underlying infrastructure or the lifecycle of the Red Hat OpenShift platform themselves. Likewise, by adopting an API-first approach at an early stage in SAP S/4HANA migration, customers can connect ROSA to the SAP backend through more generic Red Hat integration and messaging technologies, a huge upgrade on the complex dedicated integrations available with SAP's legacy integration architecture, the Enterprise Service Bus.

This approach frees organisations from having to manage a complex, infrastructure-heavy environment. It also has advantages in terms of self-service cluster creation, integration with AWS, consumption-based billing and cost saving. The whole platform is managed for the customer by a team of Red Hat engineers.

Red Hat chose AWS as its partner for ROSA to take advantage of its global hyperscaler infrastructure. This gives customers a tight integration between the underlying AWS infrastructure and features and the Red Hat OpenShift software running on top of it. In addition, AWS has a long history of partnering with SAP customers to run their workloads and in 2023 launched the AWS SDK for SAP ABAP to accelerate and simplify the integration of custom ABAP code with AWS applications. This allows developers to continue using familiar ABAP code without having to understand multiple AWS APIs.

Conclusion: the benefit of a managed platform

Eight years after SAP announced the move from SAP ECC to SAP S/4HANA, customers have had to consider numerous migration puzzles, including finding a new home for custom code and extensions, and whether to run their SAP workloads on premise, in the cloud or a mixture of the two.

The same issues confront organisations that have invested in Red Hat's Enterprise Linux® (RHEL) and the cloud native Red Hat OpenShift Kubernetes container platform. This makes an excellent home for their custom SAP extensions and custom code with easy connectivity to the SAP backend through Red Hat integrations and an API-first approach to development.

White paper

However, running on-premise Red Hat OpenShift requires expertise and a willingness to invest in the required infrastructure. For these customers, Red Hat's OpenShift Service on AWS (ROSA) offers the benefits of the Red Hat OpenShift application platform within a service jointly managed between Red Hat and AWS.

Red Hat OpenShift Service on AWS (ROSA) offers the benefits of the Red Hat OpenShift application platform within a service jointly managed between Red Hat and AWS.

This approach saves time, embraces cloud native development and most important of all, fulfills SAP's edict that the SAP S/4HANA core is kept 'clean'.

SAP's recommendation has always been a sound one – gaining the advantages of SAP S/4HANA makes it imperative that extensions are separated from the core. The question is which platform customers adopt to achieve this.

ROSA is simply another option within the Red Hat portfolio that will suit customers that want to focus on application development without managing the complex underlying infrastructure required to run on-premise Red Hat OpenShift.

Learn more about Red Hat ROSA | **redhat.com** | **aws.com**