



From Abap custom code to microservices in a hybrid cloud

Code Modernization

SAP customers have to tackle migration and modernization of their infrastructure: The switch to S/4 and Hana poses a challenge for many companies, especially considering Abap custom code.

By Peter Körner, Red Hat, and Stefan Hetges, SmartShift

With intelligent code modernization and an adequate architecture, a company can carry out complex transformation projects relatively easily, quickly, and cost-optimized. The tasks for SAP customers are clearly defined: On the one hand, they must migrate to Hana and S/4; on the other, they must simultaneously use new technologies and architectures to consistently drive modernization as part of their digital transformation - for example, with regard to the rapid and flexible introduction of new business models.

Abap custom code in legacy systems poses a significant challenge during migration. Companies that have been using SAP systems for a long time have a large amount of custom code that has been developed over years or even decades to support growth and ever-changing business requirements. This includes code for strategic business processes that companies have implemented to gain competitive advantages.

Much of this code may become obsolete over time because it is not designed with today's requirements in mind, such as security, performance, or maintainability. In addition, in many cases, the code is no longer executable in the new S/4 environment.

Abap custom code thus stands in the way of innovation by hindering modernization projects and upgrades.

Containerization

Generally speaking, the modernization process in SAP is aimed at containerizing and modifying applications to improve functions and provide upgrade support. Custom code presents a challenge as it is typically closely intertwined with the monolithic approach of traditional SAP ERP/ECC 6.0 systems. A like-to-like migration can hardly be mapped and offers no added value since extensions must also be made outside. Consequently, the tasks are to first take stock of custom code, identify ambiguous dependencies, and provide interfaces between application components. The result of these activities is increased transparency, easier maintenance, and elimination of regressions.

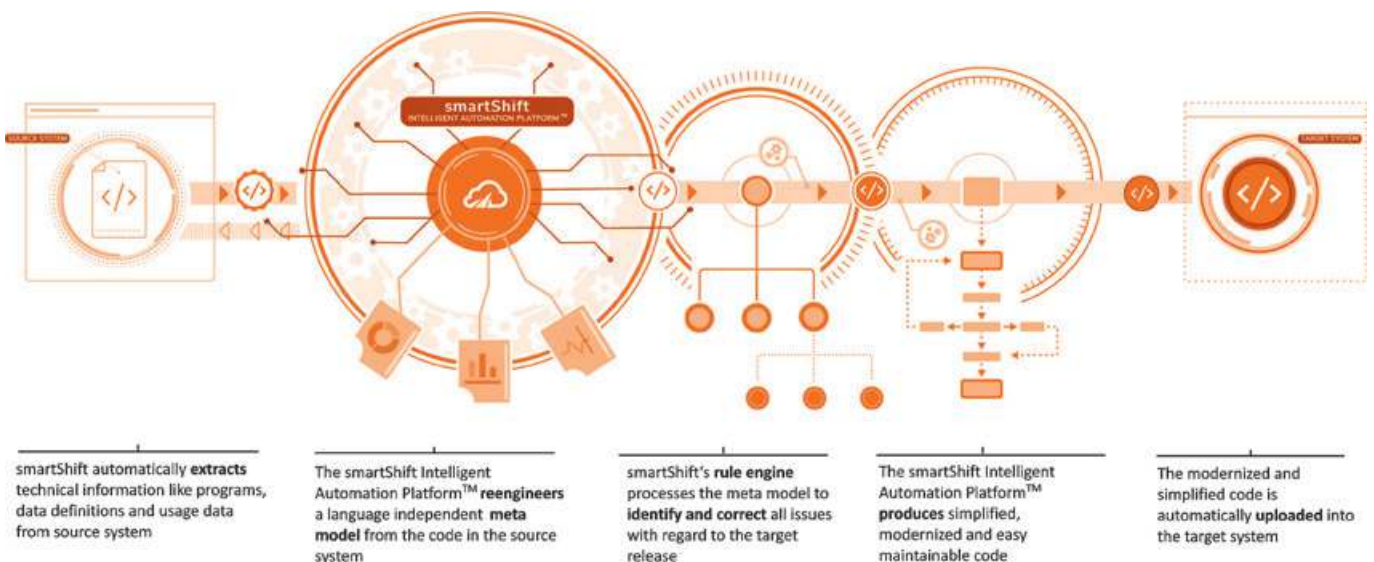
Modernizing custom code

A modernization of custom code is usually carried out in four phases: Analysis, decomposition, integration, and conception of the desired architecture.

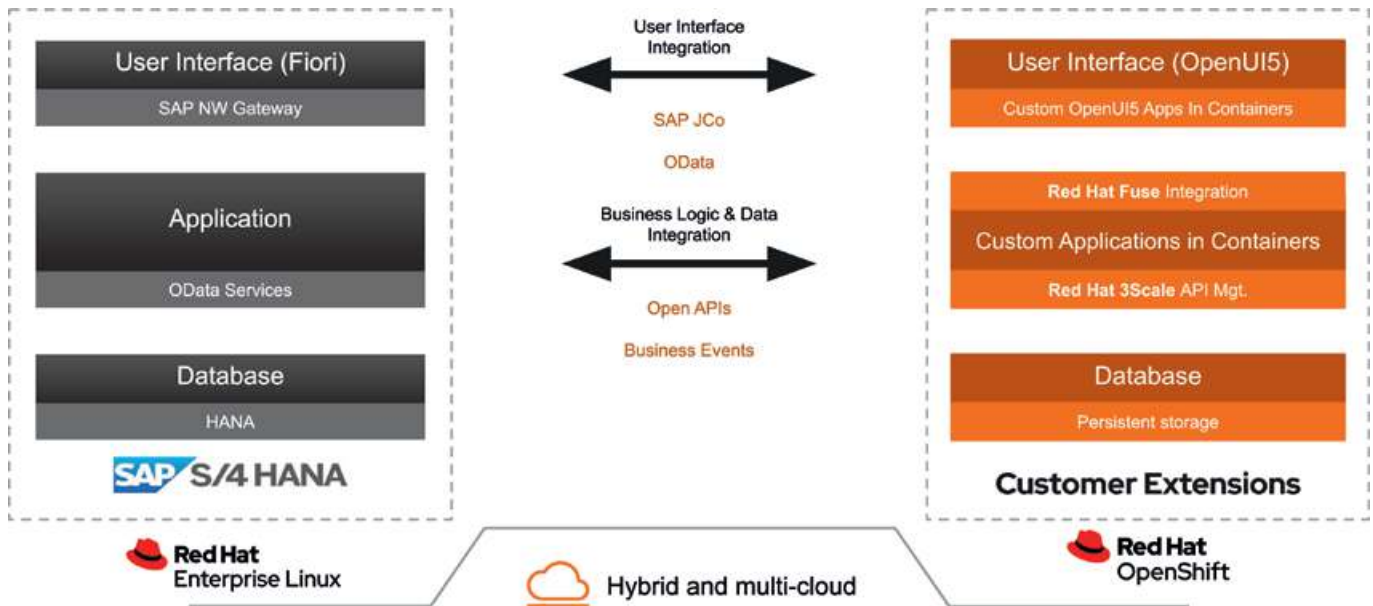
The first step involves code analysis and identifying those parts of existing system applications that are suitable for transformation into microservices and a container-based architecture. The analysis usually also reveals that there is code in SAP systems that is no longer used and can therefore be eliminated after an audit; the percentage of this unused code in a system can be up to 60 percent. The analysis identifies potential code lines that can be decoupled by showing dependencies between objects. It also identifies source code violations to make business processes executable in an S/4 environment.

Step two, the decomposition phase, involves an iterative design of components that renew the existing code base and serve as building blocks for new developments. The components support modern stateless protocol communication, so-called microservices.

The third step involves integration - connecting the components for real-time exchange of data and processes. This should involve a platform that supports loose coupling and uses API connectors, event-driven programming, and asynchronous synchronization to improve fault tolerance.



From analysis to architecture with SmartShift automation.



SAP's side-by-side extensibility concept.

The fourth and final step concerns the design of the actual architecture, which should ideally be container-based. In concrete terms, this involves, for example, covering scaling requirements or process support by means of automation regarding upgrades, monitoring, or testing.

Hybrid cloud platform

In addition to the methodology, the desired architecture is also of crucial importance for a successful migration. In the meantime, an ideal target platform has emerged.

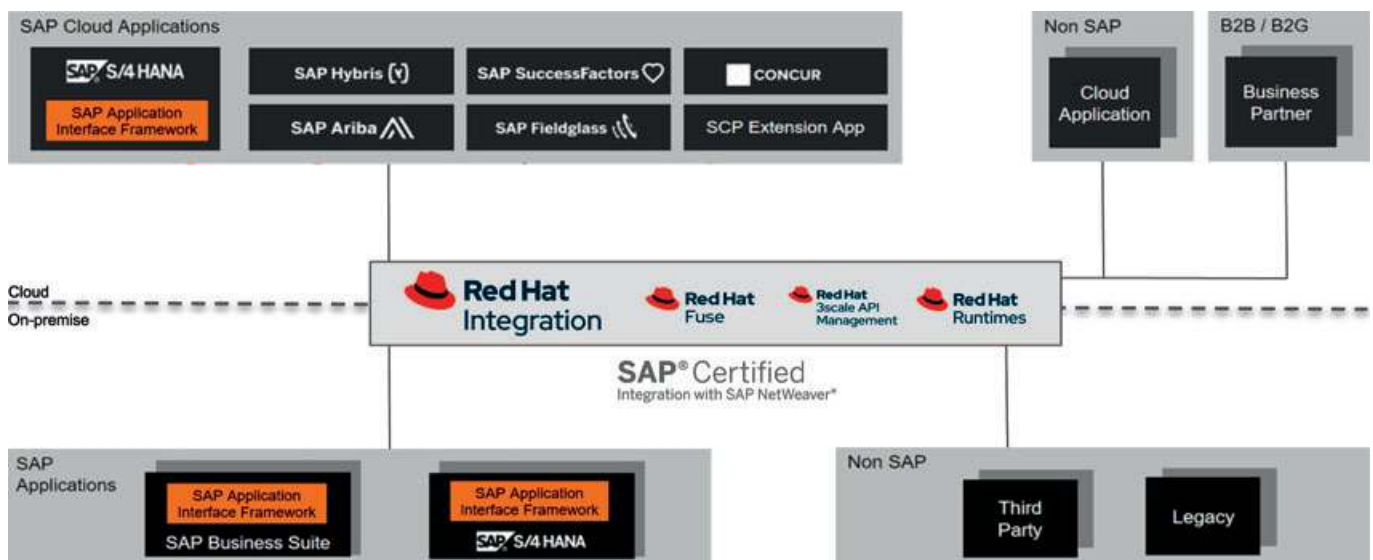
If a company wants to address innovation issues, it is clear that there is no way around the use of new platforms, frameworks, applications, and technologies in the new architecture: Integrated hybrid multi-cloud platforms, cloud-native applications, containers, microservices, and APIs

are key components here. The development is also clearly marked out in SAP: Hybrid cloud platforms and cloud-native application development will have a lasting impact on IT landscapes. The SAP ecosystem will be modernized in the direction of hybrid cloud architecture, and SAP customers will thus increasingly use a mix of on-premises, private and public cloud services in the future - not only for SAP workloads, but also for non-SAP workloads. In addition, cloud-native runtime environments will become the dominant deployment pattern, as they are the most dynamic driver of innovation.

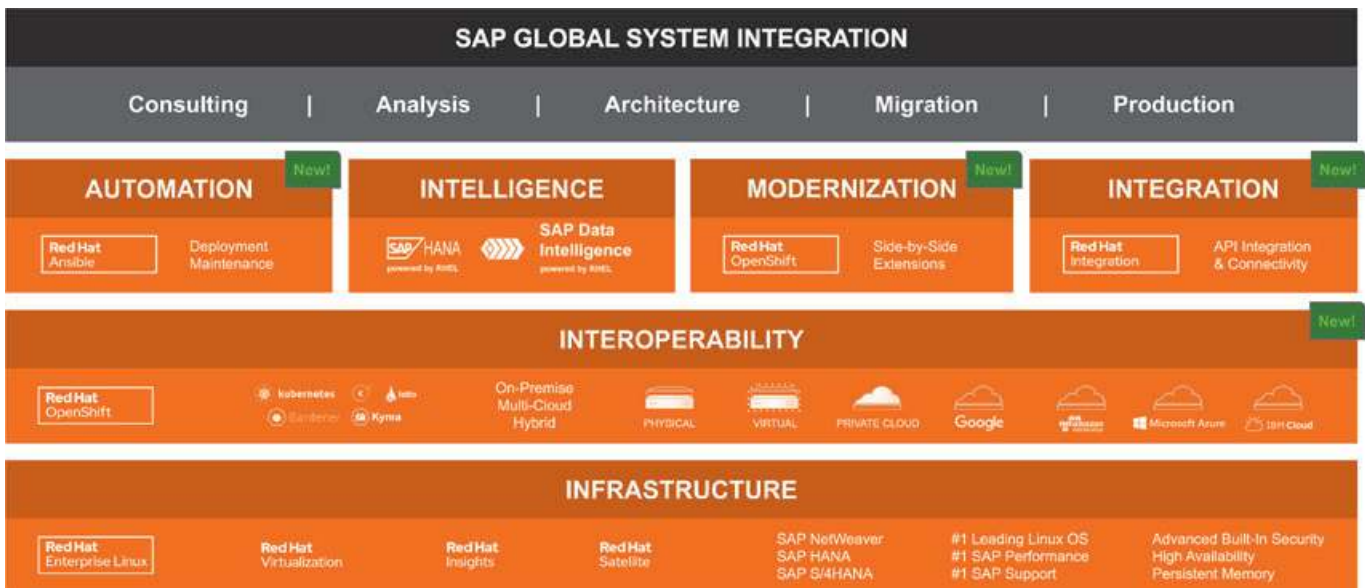
API-based communication will take on a special significance in the future SAP world. Open APIs are the technical basis for networking data, applications, and devices and thus also for implementing innovations without creating new IT silos.

Open APIs

The use of open APIs is almost indispensable, especially considering SAP's side-by-side extensibility concept. It aims to connect SAP data, processes, and user interface with modern programming environments, continuous integration and continuous delivery and other DevOps methods. So-called side-by-side extensions for S/4 systems, in contrast to classic Abap-based in-house developments, allow the simple implementation of end-to-end processes and thus also integrate the SAP landscape with non-SAP systems. This concept can be conveniently implemented on the basis of the Red Hat OpenShift enterprise Kubernetes platform; this gives users freedom of choice in terms of infrastructure and the option of multi or hybrid cloud use.



Agile integration of SAP and non-SAP workloads.



End-to-end architecture in SAP.

Integration and management solutions are needed for easy and fast implementation of the side-by-side extensibility concept. Modern API management solutions, such as those available with the enterprise version 3scale API management, are modular, highly scalable, and verifiably connectable to SAP cloud APIs and many third-party IT systems. In addition, an agile integration solution is required that enables developers to establish bidirectional communication with Hana and applications. The open source framework Apache Camel or Red Hat Fuse are particularly suitable for this purpose.

Side by side

If a side-by-side concept has been implemented, using automation solutions like Ansible becomes more convenient. Ansible supports SAP users in, for example, hy-

brid cloud deployments, meaning in the automated provisioning and management of traditional and new containerized SAP workloads as well as non-SAP applications in hybrid cloud environments. Ansible also provides DevOps automation for SAP, meaning support for automated development, deployment, go-live, and management of SAP platforms.

Overall, a new, modern hybrid cloud system architecture offers SAP users numerous advantages. These include:

- the simple and fast implementation of new requirements,
- the reduction of manual error sources through a high degree of automation,
- the rapid integration of new processes,
- the establishment of an agile development methodology,
- enabling DevOps by decoupling the end-to-end processes from the SAP core.

With the right methodology, which includes intelligent automation, and the right target platform, SAP users can perform transformations quickly with optimized risk and cost. Automation should include flexible services for analysis, performance optimization, code conversion, and consolidation. Furthermore, when selecting a platform, make sure it supports unified cloud-native application development and deployment on any infrastructure - from multi-cloud environments to on-premises deployments. Many SAP users rely on the Red Hat OpenShift enterprise Kubernetes platform, and large companies in the automotive sector, for example, have already implemented this concept in exactly the described way.



Peter Körner is Business Development Manager Open Hybrid Cloud SAP Solutions at Red Hat.



Stefan Hetges is founder and CEO of SmartShift.

Automation of code modernization

This complex SAP migration and modernization process requires a methodical approach aimed at maximum levels of automation. Automation, such as that enabled by SmartShift based on its Intelligent Automation Platform, should include the following solution components and steps with regard to SAP code analysis and transformation:

- The automatic extraction of technical information such as programs, data definitions, and usage data from the source system.
- The design of a language-independent metamodel from the code in the source system.

- A rule engine that identifies and automatically resolves any code issues in the metamodel with respect to the target version.
- A dependency analysis to understand mutual relationships among objects and to determine interfaces between domains.
- The provision of residual APIs for identified object clusters and integration with the target platform.
- The creation of simplified, modernized, and easily maintainable code.
- The automatic upload of modernized and simplified code to the target system.

Please also have a look at our community info entry

