Companies are increasingly implementing internal and external SAP and non-SAP systems that require rapid integration and stable operation. Meanwhile, the complexity of system landscapes continues to grow. Maintaining a competitive time to market and efficient operations in the face of rising SLA requirements and cost pressure thus presents a major challenge to companies and their IT organizations.

Microservices are an approach to reducing system complexity and enabling IT departments to meet new demands more quickly. For many CIOs and IT architects, they pose the following questions:

- Among all the hype surrounding the topic of microservices, what core statements stand up to scrutiny?
- What could be an optimal integration architecture with microservices?

The principles of microservices

Just as agile developers have their agile manifesto, a series of basic principles have emerged with regard to microservices:

- Separation of components through services
- Organization in specialized areas
- Products, not projects
- Smart endpoints with no middleware logic
- Decentralized governance
- Infrastructural resources (DevOps)
- Designing for failure
- Evolutionary design

These principles are the result of experiences gathered in many extensive IT projects and in landscapes that have grown large and monolithic over time. These environments are often very rigid and require redevelopment at the end of their life cycle. IT projects and changes of this magnitude are also likely to conflict with the rising level of agility necessary on the business side.

In SAP environments, it is impossible to follow these principles to the letter as in web or Java projects. Many of the underlying ideas can, however, open the door to significant improvements in a modern SAP landscape.

Insights for SAP landscapes

For years, companies ran the majority of SAP systems in on-premise configurations and conducted a great deal of custom development – especially in ERP. To realize further scenarios, they installed additional SAP systems (CRM, SRM, APO, and so on) and often integrated them using static and highly comprehensive interfaces.

When these organizations need to fulfill new requirements as quickly as possible, they now have to contend with coordination efforts, extensive dependencies, and test phases lasting up to a year before going live. Combined with the many stakeholders and projects involved, this results in highly complex and heterogeneous systems.

Problems like these can be addressed by a microservice architecture. Having self-contained services with their own specific contexts, makes it possible to quickly meet new demands. This in turn enables companies to follow a best-of-breed approach and continue advancing their IT landscape as a “product” without falling victim to excessive complexity. Specific solutions in SAP environments are illustrated on the next page.

1 http://www.agilemanifesto.org/
The trend toward splitting conventional ERP monoliths up into software components that can be used separately is also nothing new in SAP landscapes. From the Business Object Repository to enterprise services, SAP has already introduced an array of related possibilities. The company is nevertheless incorporating more of the microservice approach, as well – into its cloud offerings, for example, which are already based on the principles of this architecture. SAP Hybris Cloud (YaaS), which provides many functions through business services based on the microservice concept, deserves particular mention in this regard.

Today, users can already take advantage of this architectural approach in NetWeaver-based landscapes in order to draft manageable distributed solutions. Clearly defined APIs (specific functions from a specialized context that are encapsulated into a solution) make it easy to cover the latest requirements in a stable manner. This also reduces the need for both, in-depth ERP solution expertise and complicated efforts, to delve into the ERP system at hand, which typically depend on the software provider’s release cycles in various ways. Ideally, the solution component can be run and adjusted by a company’s in-house developer team (DevOps approach). This enables organizations to respond more flexibly to changing market conditions in critical situations.

**Implementation option #1: SAP Gateway**

In this scenario, a front-end server works as a central access point for distributed business functionality. The idea is for users to retrieve data from SAP systems by means of a Fiori app. In doing so, they can build on a variety of options, including access to other systems and implementations involving different Java-based technologies. The front-end server provides a central point of access to the various components that combine to form a business process in the Fiori app.

**Implementation option #2: in the cloud**

Through HANA Cloud Platform (HCP), SAP offers the option of using proven Java-based methods to quickly add enhanced functionality to existing SAP system landscapes. The add-ons in question can be implemented both based on HANA or by means of conventional relational-database applications. The microservices available through HCP are easy to integrate, and can also serve as a basis for new products.

Finally, those who commit to the microservice concept will need a central management tool (such as SAP API Management) in order to provide a corresponding administrative structure and oversee the life cycles of the many different microservices in play.

**About cbs**

cbs Corporate Business Solutions provides high-quality management and consulting services for SAP projects around the world. As a market-leading consultancy in SAP integration efforts and a partner of SAP SE itself, cbs maintains close ties to the company’s production development division.

cbs Corporate Business Solutions Unternehmensberatung GmbH
Denny Schreber · Senior Solution Architect · denny.schreber@cbs-consulting.de
Thorsten Düvelmeyer · Senior Solution Architect · thorsten.duevelmeyer@cbs-consulting.de

Im Breitspiel 19 · 69126 Heidelberg, Germany · T +49 (0) 6221 3304-0 · www.cbs-consulting.com