

Kapitel 2

SAP S/4HANA – die technische Konzeption

SAP HANA ist eine tragende Säule von SAP S/4HANA. Sie ermöglicht es, Daten in Echtzeit zu analysieren. Mit der SAP Cloud Platform eröffnen sich völlig neue Möglichkeiten für die Programmierung und Erweiterung des Systems. Dieses Kapitel erklärt Ihnen diese Änderungen und weitere technische Grundlagen von SAP S/4HANA.

In Kapitel 1, »Konzepte von SAP S/4HANA«, haben wir Ihnen die Anforderungen an moderne IT-Systeme vor dem Hintergrund des digitalen Wandels vorgestellt. Wir haben Ihnen gezeigt, welche Lösungsansätze SAP vor diesem Hintergrund bei der Entwicklung von SAP S/4HANA gewählt hat. In diesem Kapitel wollen wir Ihnen nun die technischen Komponenten und Prinzipien von SAP S/4HANA detaillierter erklären.

In Abschnitt 2.1, »Die SAP HANA Plattform«, nehmen wir die Datenbank SAP HANA als technische Basis in den Blick. Sie eröffnet Ihnen mithilfe der In-Memory-Technologie ganz neue Ansätze für die Entwicklung von SAP-Anwendungen.

Abschnitt 2.2, »Entwicklung unter SAP S/4HANA«, stellt die Möglichkeiten in den Vordergrund, die SAP S/4HANA für die Entwicklung von Anwendungen bereitstellt: Geänderte Paradigmen bedeuten große Veränderungen für SAP-Entwickler. Um die Möglichkeiten von SAP HANA ausschöpfen zu können, werden für die Entwicklung Komponenten aus dem Applikationsserver in die Datenbank verlagert. Diese wiederum sind mit der sogenannten ABAP Workbench – der Standardentwicklungsumgebung im ABAP-Umfeld – nicht zu bewerkstelligen. Aus diesem Grund werden neue Werkzeuge wie das SAP HANA Studio eingesetzt. Dabei werden sowohl neue Entwicklungsoptionen auf der SAP-HANA-Datenbank als auch Erweiterungen von Open SQL unter ABAP unterstützt.

Zur Auswertung von Daten in SAP HANA stehen unterschiedliche Werkzeuge zur Verfügung: In Abschnitt 2.3 beschreiben wir Werkzeuge, die eine Datenanalyse in Echtzeit direkt in SAP S/4HANA ermöglichen. Es ist dann z. B. nicht mehr notwendig, einen Umweg über SAP Business Warehouse

(SAP BW) zu gehen. In diesem Zusammenhang stellen wir Ihnen SAP Lumira und SAP Analysis vor.

Die neue Benutzeroberfläche gehört ebenfalls zu den zentralen Bausteinen von SAP S/4HANA (siehe Abschnitt 2.4, »SAP Fiori«). Wir gehen detailliert auf die Technologien und Prinzipien ein, die hinter den Fiori-Apps stehen.

In Abschnitt 2.5 stellen wir die SAP Cloud Platform in den Mittelpunkt. Dieses Platform-as-a-Service-(PaaS-)Angebot von SAP bietet Ihnen eine Vielzahl cloudbasierter Dienste, mit denen das SAP-S/4HANA-System erweitert werden kann. Die SAP Cloud Platform dient dabei z. B. als Laufzeitumgebung für eigene Anwendungen oder stellt cloudbasierte Entwicklungstools wie die SAP Web IDE zur Verfügung.

Machine Learning, künstliche Intelligenz und Robotik sind wesentliche Funktionen im Kontext von SAP S/4HANA – in Abschnitt 2.6, »Künstliche Intelligenz«, stellen wir Ihnen vor, welche Konzepte für die Einbettung von Machine Learning in die Lösung bestehen, erläutern an Beispielen Anwendungsszenarien und stellen auch dar, wie mit Robotic Process Automation Abläufe in SAP S/4HANA automatisiert werden können.

2.1 Die SAP HANA Plattform

Der zentrale Kern von SAP S/4HANA ist die Datenbank SAP HANA. Diese macht sich verschiedene Techniken zunutze, um die Auswertungen großer Datenmengen mit hoher Geschwindigkeit zu ermöglichen. Die wichtigsten Konzepte in diesem Kontext sind die *In-Memory-Technologie* und die *Spaltenorientierung*, die in Abschnitt 2.1.1 und Abschnitt 2.1.2 erläutert werden. Sie ändern fundamental die Art und Weise, wie das Datenbanksystem mit den zu verarbeitenden Daten umgeht und sind somit auch wesentlich für das Verständnis der anwendungsbezogenen Veränderungen in SAP S/4HANA, die wir in späteren Kapiteln dieses Buches thematisieren.

Da es in spaltenbasierten Datenbanktabellen vergleichsweise aufwendig sein kann, neue Datensätze einzufügen oder zu ändern, wurde in SAP HANA das sogenannte *Delta-Merge-Verfahren* umgesetzt. Es verhindert, dass aktive Leseprozesse von den langsameren Schreibprozessen behindert werden. Wir beschreiben dieses Konzept in Abschnitt 2.1.3.

In Abschnitt 2.1.4 zeigen wir dann, inwiefern SAP HANA das aus der Datenbanktheorie bekannte *ACID-Prinzip* (Atomicity, Consistency, Isolation, Durability) einhält. In Abschnitt 2.1.5 erfahren Sie, welche Architektur der Datenbank zugrunde liegt.

Im Kontext der Datenbank SAP HANA bietet SAP eine Vielzahl an zusätzlichen Funktionen und Diensten an. Diese Gesamtheit aus Datenbank und Funktionen/Diensten bildet in Summe die SAP HANA Plattform. Wir geben in Abschnitt 2.1.6, »Services auf der SAP HANA Plattform«, einen Überblick über diese Dienste, um deutlich zu machen, dass SAP HANA noch deutlich mehr bietet, als »nur« eine gesteigerte Performance oder die Datenbankfunktionalität.

2.1.1 In-Memory-Technologie

Die Datenbank SAP HANA basiert vollständig auf der *In-Memory-Technologie*. Das bedeutet, dass ein Großteil aller operativ genutzten Daten im Arbeitsspeicher gehalten und dort verarbeitet wird. Dies unterscheidet SAP HANA von herkömmlichen Datenbankmanagementsystemen, die die Daten auf Festplattenlaufwerken speichern. Der wesentliche Vorteil liegt in den deutlich besseren Zugriffszeiten des Arbeitsspeichers, da dieser in der Rechnerarchitektur näher an den verarbeitenden Prozessoren liegt. Ein Nachteil ist hingegen, dass die Kosten des Arbeitsspeichers höher sind. Durch den andauernden Preisverfall und die stetig steigende Speicherkapazität ist dieser Nachteil aber in den letzten Jahren immer kleiner geworden. Zudem werden in einer SAP-HANA-Datenbank Techniken zur Datenkompression (z. B. Dictionary Encoding) verwendet, die dafür sorgen, dass mit dem vorhandenen Arbeitsspeicher möglichst effizient umgegangen wird.

Die Verwendung des In-Memory-Konzepts führt zu einer minimalen Latenz bei der Verarbeitung der Daten durch den Prozessor. Antwortzeiten können so deutlich verkürzt und Auswertungen und Prozesse signifikant beschleunigt werden. In SAP S/4HANA stellt die In-Memory-Datenbank die primäre Persistenz dar; die »physische Datenbank« wird hierbei vollständig ersetzt und in den Arbeitsspeicher verlagert. Lediglich zu Archivierungszwecken und für Wiederherstellungsszenarien erfolgt eine physische Sicherung der Daten.

2.1.2 Spalten- und zeilenbasierte Datenbanken

Alle Daten können aus logischer Sicht als zweidimensionale Datenstrukturen betrachtet werden, die aus Zeilen und Spalten bestehen. Auf der physischen Schicht der Datenablage im speichernden Medium werden diese Datenstrukturen allerdings als eine eindimensionale Folge von Bytes abgelegt und verwaltet.

In-Memory-
Technologie

Zeilenbasierte Datenbanken

Herkömmliche Datenbankarchitekturen verwalten ihre Daten in der Regel *zeilenbasiert*. Hierbei werden die einzelnen Elemente eines Datensatzes als aufeinanderfolgender Bytestring gespeichert. Alle Daten eines Datensatzes liegen bei diesem Ansatz nebeneinander im Speicher, was den Zugriff auf »ganze« Datensätze optimiert.

Schwieriger gestaltet sich bei diesem Ansatz der gezielte Zugriff auf eine begrenzte Teilmenge an Informationen eines Datensatzes (z. B. auf den Saldo eines Sachkontos der FI-Einzelposten (Tabelle BSEG). Hierzu muss der Bytestring zerlegt werden, um die relevanten Informationen auszulesen.

Dies führt häufig dazu, dass eine gesamte Bytefolge in die Applikationsebene geladen wird und vom System aufgelöst und verarbeitet werden muss. Dieser Ansatz führt zu längeren Antwortzeiten, einem erhöhten Datenvolumen auf den Applikationsebenen und somit zu einem erhöhten Zeit- und Ressourcenbedarf des Systems.

Diese Aspekte schlagen sich in längeren Antwortzeiten einer jeden Systemoperation nieder. Bei Einfüge-, Änderungs- und Löschoptionen, die einen ganzen Datensatz betreffen, hat dieses Konzept aber Vorteile, da hier ein zusammenhängender Speicherblock adressiert werden kann.

Spaltenbasierte Datenbanken

Mit SAP HANA greift SAP auf den Ansatz der physisch *spaltenbasierten* Datenhaltung zurück. Hierbei werden die Inhalte der einzelnen Spaltenelemente einer logischen Datensicht als Bytefolge abgelegt.

Der logische Aufbau eines Datensatzes wird also im Rahmen des Persistierens aufgelöst. Dieses Vorgehen hat zur Folge, dass Datenoperationen, die nur auf ausgewählte Spalten einer logischen Tabelle zugreifen, deutlich schneller ausgeführt werden können, als es ein zeilenbasierter Ansatz ermöglichen würde. So können z. B. Summen über ausgewählte Attribute deutlich schneller berechnet werden.

Ebenso reduzieren sich der Ressourcenbedarf und der Verarbeitungsaufwand der Daten in den Applikationsebenen, da jetzt gezielt nur die Daten geladen werden, die für die einzelnen Operationen und Prozesse auch erforderlich sind.

Zugriffe, die auf komplette logische Datensätze erfolgen, führen in diesem Fall allerdings zu einem erhöhten Verarbeitungsaufwand: In diesem Fall müssen die einzelnen Datensätze wieder komplett als logischer Datensatz zusammengeführt werden.

Abbildung 2.1 illustriert den Unterschied zwischen der zeilenbasierten und der spaltenbasierten physischen Datenablage. Während bei der zeilenbasierten Speicherung die Zeileninhalte nacheinander abgelegt werden, sieht die spaltenbasierte Ablage zunächst eine Ablage der Spaltenwerte aus Spalte 1,

dann aus Spalte 2 usw. vor. Einfüge-, Änderungs- und Löschoptionen sind im Kontext einer spaltenbasierten Datenablage ressourcenintensiv.

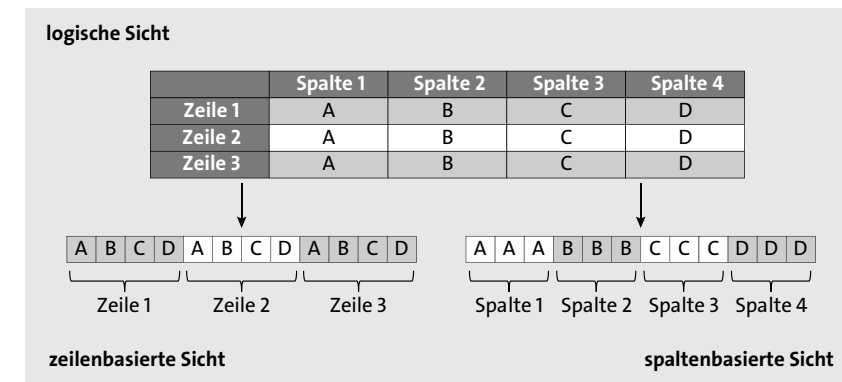


Abbildung 2.1 Zeilen- und spaltenbasierte Datenablage

2.1.3 Delta-Merge-Verfahren

Einfüge-, Änderungs- und Löschoptionen werden aufgrund der ressourcenintensiven Datenbankprozesse nach dem sogenannten *Delta-Merge-Verfahren* durchgeführt. Dieses Verfahren soll sicherstellen, dass mögliche aktive Leseoperationen nicht von den langsameren Schreibprozessen behindert und verlangsamt werden.

Das Delta-Merge-Verfahren sammelt alle durchgeführten Änderungen in einem schreiboptimierten Delta-Storage (pro Datenbanktabelle) und speichert diese dort zwischen. Zeitgleich werden die betroffenen Einträge im Main-Storage als überarbeitet und somit als ungültig markiert. Alle Leseoperationen greifen zu diesem Zeitpunkt sowohl auf den Main-Storage als auch auf die aktuelleren Delta-Storages zu. So ist sichergestellt, dass Ihre Datenzugriffe immer auf dem aktuellen Datenstand basieren.

Durch dieses Vorgehen soll das System entlastet werden und soll die bestmögliche Performance der lesenden Prozesse sichergestellt werden, da die Transformation der veränderten Daten in die speicherplatz- und lesezugriffsoptimierte spaltenbasierte Datenablage sehr ressourcenintensiv ist. Die Umwandlung der veränderten Daten erfolgt automatisiert zu einem späteren Zeitpunkt, an dem die Systemauslastung gering ist.

Abbildung 2.2 zeigt den schematischen Ablauf des Delta-Merge-Verfahrens. Während der Durchführung des Delta-Merge-Verfahrens werden temporär neue Main-Storages für die betroffenen Tabellen angelegt. Diese temporären Main-Storages ersetzen nach Abschluss des Delta-Merge-Verfahrens die ursprünglichen und zu aktualisierenden Main-Storages. Zusätzlich werden

Delta-Storage und Main-Storage

mit Beginn des Verfahrens neue Delta-Storages erzeugt. Das Delta-Merge-Verfahren überführt alle Änderungen aus den Delta-Storages in die Main-Storages und baut die alten Delta-Storages im Anschluss ab.

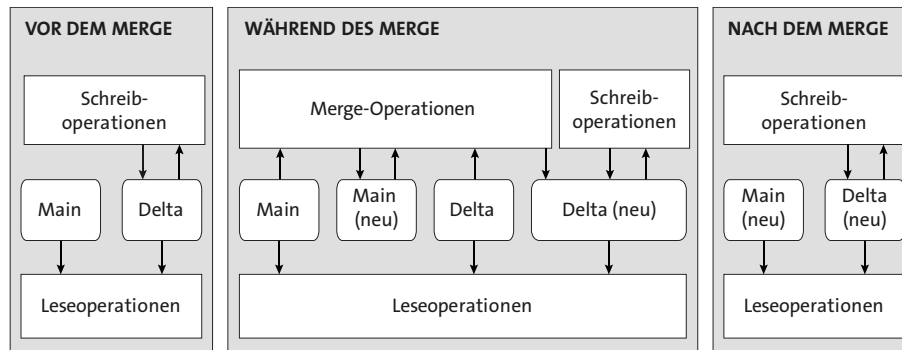


Abbildung 2.2 Delta-Merge-Verfahren

2.1.4 Das ACID-Prinzip

Bestandteile des ACID-Prinzips

SAP HANA garantiert als transaktionale Datenbank die Einhaltung der vier grundlegenden Eigenschaften des sogenannten *ACID-Prinzips*. Diese sind:

- **Atomarität (Atomicity)**

Jede durchgeführte Transaktion eines Anwenders muss atomar erfolgen. Dies bedeutet, dass sie entweder vollständig oder überhaupt nicht ausgeführt wird. Das Prinzip der Atomarität betrifft im Speziellen alle schreibenden Datenbankprozesse einer Transaktion. Tritt im Verlauf einer Transaktion ein Fehler (Software- oder Hardwarefehler) auf, setzt SAP HANA alle bereits ausgeführten Datenbankoperationen zurück (Datenbank-Rollback) und entfernt die Daten vollständig von der Datenbank.

- **Konsistenz (Consistency)**

Die Datenbankkonsistenz vor und nach der Durchführung einer Transaktion ist gewährleistet.

- **Isolation (Isolation)**

SAP HANA unterstützt das parallele Arbeiten einer großen Anzahl von Anwendern. Die Datenbank stellt durch geeignete Sperrverfahren sicher, dass aus dem Mehrbenutzerbetrieb heraus keine inkonsistenten oder nicht nachvollziehbaren Datenstände resultieren.

- **Dauerhaftigkeit (Durability)**

SAP HANA garantiert die Dauerhaftigkeit der Daten. Datenbankoperationen von abgeschlossenen Transaktionen überleben alle möglichen zukünftigen Fehler des Systems und sind im schlimmsten Fall vollständig wiederherstellbar.

Speziell der Aspekt der Dauerhaftigkeit ist bei In-Memory-Datenbanken besonders relevant, da bei einem Systemcrash die Daten des Arbeitsspeichers verloren gehen können. Der Arbeitsspeicher, als führendes SAP-HANA-Medium, ist ein volatiles Speichermedium, dessen Inhalt z. B. bei einem Stromausfall vollständig verloren geht.

Das Sichern der Daten einer SAP-HANA-Datenbank erfolgt deshalb in regelmäßigen Abständen über definierbare *Sicherungspunkte*. Beim Erreichen eines Sicherungspunktes werden die Daten auf physische Speichersubsysteme übertragen und dort persistiert. Die Datenübertragung erfolgt hierbei parallel zum normalen operativen Systembetrieb und führt hier zu keinerlei Beeinträchtigungen.

Im Zeitraum zwischen zwei Sicherungspunkten wird zusätzlich ein *Transaktionslog* geführt, das die Ergebnisse aller atomar abgeschlossenen Transaktionen und der zugehörigen Datenbankoperationen protokolliert. Dieses Transaktionslog wird laufend auf systemnahen Flashspeichern gesichert.

Die Kombination aus Sicherungspunkten und Transaktionslog garantiert die Konsistenz und Dauerhaftigkeit der Daten auch nach einem vollständigen Systemausfall und erlaubt den Wiederaufbau eines Systemzustandes, wie in Abbildung 2.3 dargestellt.

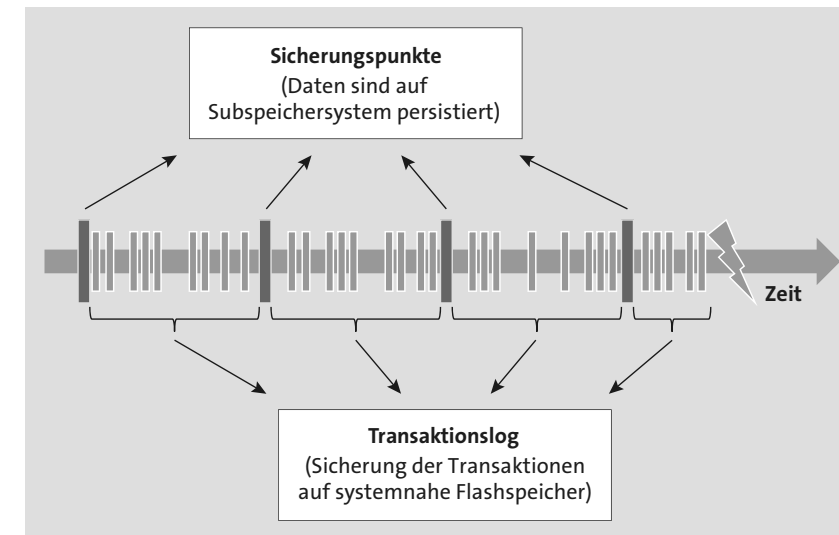


Abbildung 2.3 Sicherstellung der Dauerhaftigkeit

Anhand der Daten des letzten Sicherungspunktes kann nach einem Abbruch das System wieder aufgebaut werden. Alle Transaktionen (bzw. deren Datenbankoperationen), die nach dem letzten Sicherungspunkt erfolgreich

beendet wurden, werden im Anschluss erneut durchgeführt. So wird das System wieder in den letzten vorliegenden stabilen Zustand versetzt.

2.1.5 Technischer Aufbau von SAP HANA

Server von SAP HANA Die SAP-HANA-Datenbank besteht aus den folgenden fünf wesentlichen Servern:

- **Index Server**
Der Index Server ist die zentrale Serverkomponente, die die aktuellen Daten sowie die Zugriffe auf die Daten steuert und verwaltet.
- **Name Server**
Der Name Server verwaltet Informationen zur Topologie der SAP-HANA-Datenbank. Bei verteilten Systemen verwaltet er zusätzlich die Informationen, wo welche Dienste und Funktionen laufen und wo welche Daten abgelegt sind.
- **Statistics Server**
Der Statistikserver verwaltet die Daten über den Status, die Performance und den Ressourcenverbrauch aller SAP-HANA-Komponenten. Er verfügt außerdem über Werkzeuge für das Monitoring der SAP-HANA-Datenbank.
- **Preprocessor Server**
Der Preprocessor Server wird vom Index Server genutzt, um spezielle Funktionen zur Textsuche und -analyse durchzuführen.
- **XS Engine**
Die SAP HANA XS Engine ist ein Applikationsserver, der den Zugriff via HTTP erlaubt. Diese Komponente ist optional.

Abbildung 2.4 illustriert die erläuterten Server der SAP-HANA-Datenbank. Für den Index Server, der die Kernkomponente darstellt, nimmt die Grafik eine weitere Unterteilung vor, die wir im Folgenden kurz darstellen.

Komponenten des Index Servers Der Index Server verwaltet die Komponenten, die für die grundlegenden Datenbankzugriffe verantwortlich sind:

- **SQL-Prozessor**
Der SQL-Prozessor führt die schreibenden Datenbankzugriffe durch und leitet alle anderen Zugriffe an die relevanten Komponenten weiter.
- **SQLScript**
SQLScript ist die SAP-HANA-eigene Datenbanksprache. Ihr Fokus liegt auf der Optimierung und Parallelisierung von Datenbankabfragen. Sie stellt eine Sammlung von Erweiterungen zur SQL-Standardsyntax dar.

- **MDX (Multidimensional Expressions)**
MDX ist eine Datenabfragesyntax zum Auslesen und Arbeiten mit multidimensionalen Datenstrukturen. Diese Syntax ist bisher weitgehend im Bereich der OLAP-Cubes von SAP BW verwendet worden.
- **Planungs-Engine**
Die Planungs-Engine führt alle Planungsoperationen (z. B. zur Finanzplanung) aus. Hierzu gehören z. B. Funktionen zum Anlegen neuer Planungsversionen.
- **Berechnungs-Engine**
Die Berechnungs-Engine verwaltet die zentralen Berechnungslogiken, die bei Datenbankabfragen relevant sind. SQLScript, MDX-Abfragen und Planungsfunktionen werden bei ihrer Ausführung in die hier verarbeiteten Berechnungsmodelle überführt und dann verarbeitet.

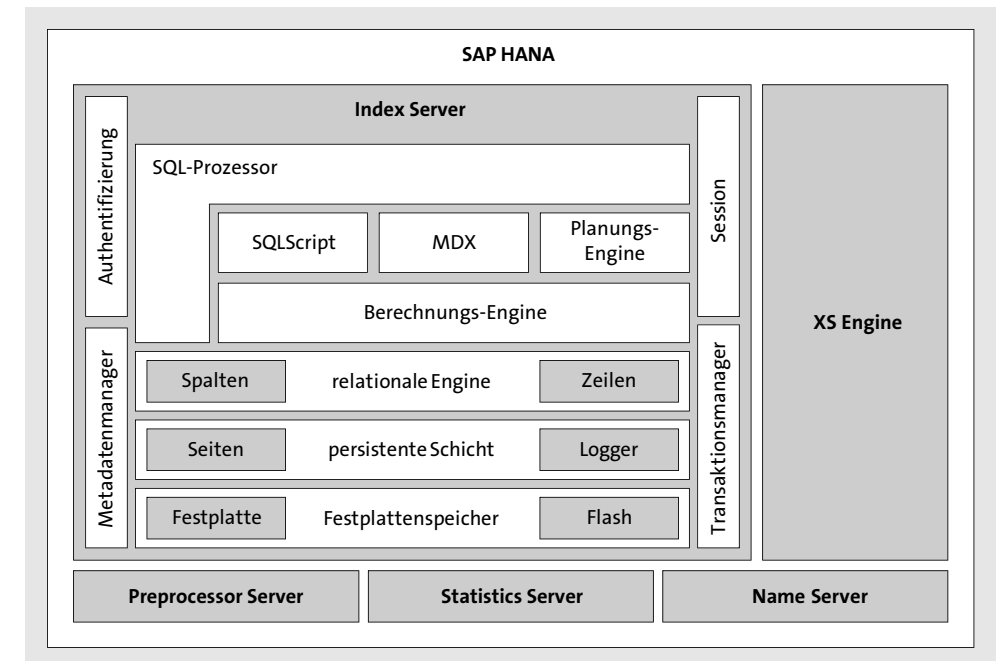


Abbildung 2.4 Architektur von SAP HANA

2.1.6 Services auf der SAP HANA Plattform

Bisher haben wir SAP HANA vor dem Hintergrund der Kernaufgaben eines Datenbanksystems, der Speicherung und Bereitstellung von Daten beschrieben. Die SAP HANA Plattform bietet darüber hinaus aber viele weitere Funktionen und Dienste, die auf diesen Grundfunktionalitäten aufsetzen.

Viele Services auf der SAP HANA Plattform

Das Spektrum reicht hier von eher klassischen Funktionen zur Datenbank-administration über Werkzeuge zur Textsuche und -analyse bis hin zu sehr innovativen Funktionen aus dem Bereich der vorausschauenden Analyse (Predictive Analysis).

Es gibt zwei verschiedene Editionen, in denen Sie SAP HANA nutzen können. Die *SAP HANA Standard Edition* beinhaltet insbesondere Datenbank-services, Integrationsdienste und Anwendungsdienste. Die *SAP HANA Enterprise Edition* bietet zusätzlich zu allen Funktionen der Standardedition weitere Dienste, Services und Funktionen.

Es ist nicht der Anspruch dieses Buches, alle Funktionen, die die SAP HANA Plattform bietet, erschöpfend darzustellen. Stattdessen sollen im Folgenden ausgewählte Aspekte aus verschiedenen Bereichen beispielhaft genannt werden, um Ihnen einen Eindruck von der Bandbreite verfügbarer Dienste und Funktionen zu geben.

Datenbankservices

Neues Konzept der Anwendungs-entwicklung

Neben den grundlegenden In-Memory-Datenbank-Services, die wir in den vorausgehenden Abschnitten beschrieben haben, bietet SAP HANA neue Möglichkeiten der Datenmodellierung. Mit der nativen Programmiersprache SQLScript ist es möglich, Datenbankzugriffe kombiniert mit Verarbeitungslogik in Prozeduren zu kapseln und direkt auf der Datenbank auszuführen. Über die Definition von sogenannten *Core Data Services Views* (CDS Views) können komplexe Sichten umgesetzt werden. Diese Konzepte beschreiben wir im Detail im nachfolgenden Abschnitt 2.2, »Entwicklung unter SAP S/4HANA«.

Analyseprozesse

SAP HANA bietet Unterstützung für verschiedene Analyseverfahren.

Spezielle Sprachkonstrukte für die Datenanalyse

Für die Analyse umfangreicher Reihendaten (z. B. den Stromverbrauch einer Maschine im Zeitablauf) bietet SAP HANA spezielle Sprachkonstrukte zur Definition und möglichst effizienten Verarbeitung und Auswertung (z. B. Ausreißeridentifikation, Aggregation, Regression) von Reihendatentabellen bei gleichzeitiger Erreichung einer hohen Datenkompression.

Streaming-Analyse

Die Streaming-Analyse verbindet eingehende Datenströme (z. B. Preisentwicklungen in Finanzmärkten) mit den in der SAP-HANA-Datenbank gespeicherten Geschäftsdaten. Die Datenströme können bezüglich Trends und Korrelationen ausgewertet werden und automatisiert Aktionen (z. B. die Benachrichtigung einer verantwortlichen Person) auslösen. Es wird ein separater Streaming Analytics Server benötigt, der mit dem SAP-HANA-Datenbankserver verbunden wird.

Die Textanalysefunktionen in SAP HANA dienen dem Zweck, unstrukturierte textbasierte Daten in strukturierte (und somit auswertbare) Daten zu überführen. Zum einen kann nach Stichworten im Textbestand der SAP-HANA-Datenbank gesucht (Volltext- und unscharfe Suche), zum anderen können komplexe linguistische und statistische Techniken genutzt werden, um aus dem unstrukturierten Text Kerninformationsobjekte zu extrahieren und z. B. für eine Stimmungsanalyse zugänglich zu machen. Mithilfe von Text-Mining-Algorithmen können zudem Muster in den Texten erkannt werden.

Mithilfe der Geodaten-Engine in SAP HANA können räumliche Daten effizient gespeichert und verarbeitet werden. Hierfür stehen eigene Datentypen und Funktionen bereit.

Anwendungsentwicklung

Mit den SAP HANA Extended Applications Services (SAP HANA XS) bringt SAP HANA einen eigenen, reduzierten Anwendungs- und Webserver mit (siehe Abschnitt 2.1.5, »Technischer Aufbau von SAP HANA«). Dieser ermöglicht es, ohne zusätzliche architektonische Ebenen (und somit sehr performant) Anwendungen auf Basis von SAP HANA zu entwickeln und auszuführen. Die Idee ist hierbei, rechenintensive, datenbankbezogene Logik direkt auf der SAP-HANA-Datenbank (z. B. mittels SQLScript) abzubilden und die Frontend-Berechnungen bzw. die Aufbereitung der Benutzermasken im Webbrowser (mittels HTML5 und JavaScript) durchzuführen. Die Anwendungsschicht des XS kümmert sich in diesem Szenario somit »nur« um die Ablaufsteuerung und nutzt hierzu Technologien wie serverseitiges JavaScript oder OData (Open Data Protocol).

Datenzugriff

SAP HANA bietet Funktionen zur Echtzeitdatenreplikation und Datenintegration. Mittels sogenannter Adapter können Daten aus Quelldatenbanken (z. B. Apache Hadoop) geladen werden. Eigene Adapter können kundenindividuell entwickelt werden.

Durch Datenbündelung (SAP HANA Smart Data Access) ist es in SAP HANA möglich, Datenbestände aus anderen Datenbanken als »virtuelle Tabellen« zu integrieren. Auf diese virtuellen Tabellen kann wie auf konventionelle Tabellen lesend und schreibend zugegriffen werden. Auf diese Weise kann gegebenenfalls auf eine aufwendige Datenreplikation zwischen den Systemen verzichtet werden. Unterstützte Datenbanken sind beispielsweise Microsoft SQL Server 2012, Oracle Database 12C, IBM DB2 und viele weitere.

Überführung von unstrukturiertem Text in strukturierte Daten

Speicherung von Geodaten

Echtzeitdatenreplikation und Datenintegration

Werkzeuge zur Administration**Administration**

In SAP HANA sind viele Werkzeuge zur Administration und zum System-Monitoring vorhanden. Mit dem SAP HANA Cockpit kann beispielsweise eine einzelne SAP-HANA-Datenbank webbasiert administriert werden, während das SAP DB Control Center aggregierte Informationen zu mehreren SAP-HANA-Instanzen liefert.

Für das Testen neuer/angepasster Systemkonfigurationen bietet SAP HANA ein Werkzeug, das die Arbeitslast eines realen Produktivsystems automatisch aufzeichnen und diese in einer Testumgebung abspielen kann (Capture and Replay Performance Management).

SAP HANA unterstützt ein breites Spektrum an Techniken, die der Umsetzung von Hochverfügbarkeit und Disaster Recovery dienen. Dazu zählen beispielsweise die Konzepte zur Systemreplikation und Back-up-Mechanismen.

IT-Sicherheit**Verschlüsselungs- und Prüftechniken**

Die Datenverarbeitung und Kommunikation innerhalb der SAP HANA Plattform kann mittels Verschlüsselungs- und Prüftechniken abgesichert werden. Es werden zudem verschiedene Techniken zur Realisierung von Single Sign-on (SSO) unterstützt.

2.2 Entwicklung unter SAP S/4HANA

Mit dem Einsatz von SAP HANA sowie der geschilderten In-Memory-Technologie ergeben sich völlig neue Möglichkeiten, sowohl einerseits für die Entwicklung neuer Anwendungen als auch andererseits für die Erweiterung von bestehenden Anwendungen. Um dieses neue Potenzial jedoch ausschöpfen zu können, muss sich auch die Vorgehensweise bei der Entwicklung von SAP-Anwendungen wandeln.

Vom Applikationsserver zur Datenbank

Open SQL bietet bei der Entwicklung von ABAP-Programmen seit vielen Jahren eine Abstraktionsebene im Hinblick auf die verwendete Datenbank. Aufgrund der Unabhängigkeit von der eigentlichen Datenbank mussten sich die Entwickler lange Zeit keine Gedanken über Hersteller oder Version machen. Viele Entwickler haben sich dadurch einen Programmierstil angeeignet, der den Fokus sehr auf den Applikationsserver und weniger auf die Datenbank legt. Häufig hatte dies zur Folge, dass große Datenmengen von der Datenbank in den Applikationsserver gelesen und dort verarbeitet wurden.

Im Zuge des digitalen Wandels und weil die Datenmengen so enorm wachsen (z. B. durch das Internet of Things und soziale Medien, siehe Kapitel 1, »Konzepte von SAP S/4HANA«), ist diese Vorgehensweise bei der Entwicklung zukünftig nicht mehr praktikabel. Es dauert schlicht viel zu lange, derartige Datenmengen aus der Datenbank zunächst in den Applikationsserver zu laden und sie dort zu verarbeiten.

Bei der Antwort auf diese Herausforderung kann die SAP-HANA-Datenbank ihre Stärken ausspielen. Die In-Memory-Technologie von SAP HANA erlaubt es Entwicklern, rechenintensive Prozesse direkt in die Datenbank zu verlagern. Auf diese Weise entfällt die Übertragung von großen Datenmengen an den Applikationsserver, die früher notwendig war. Es entsteht ein enormer Geschwindigkeitsgewinn. Diese Verlagerung der Berechnung wird als *Code Pushdown* bezeichnet.

Konzept des Code Pushdowns

Abbildung 2.5 soll Ihnen diese Verschiebung von Rechenoperationen veranschaulichen: Während diese Operationen in der klassischen Entwicklung noch in der Applikationsschicht verankert waren, werden sie unter SAP HANA so weit wie möglich in die Datenbankschicht verschoben.

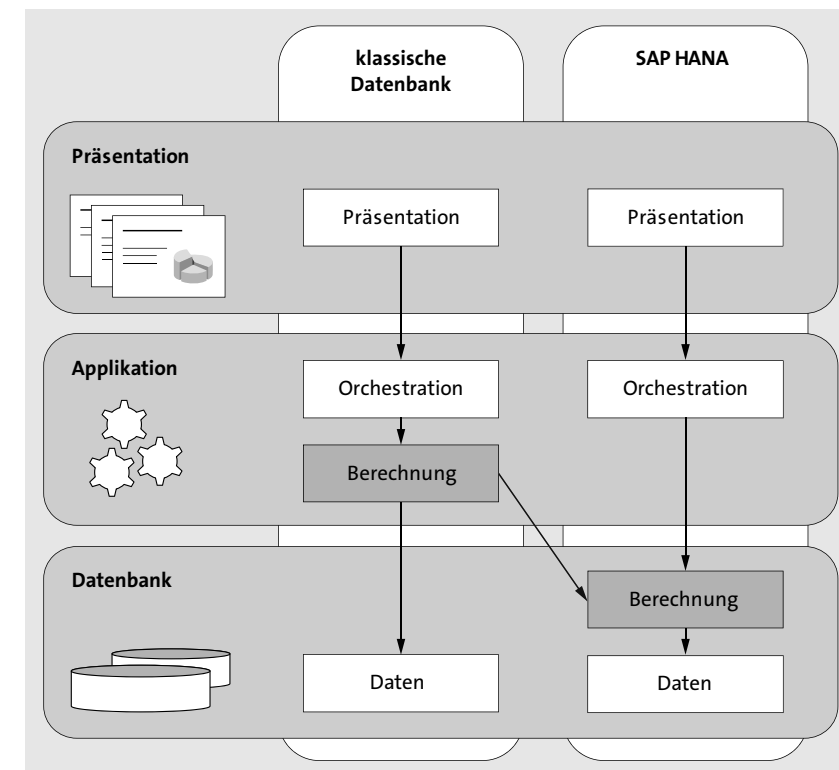


Abbildung 2.5 Code Pushdown

Beim Code Pushdown geht es also darum, rechenintensive Kalkulationen aus der Applikationsschicht in die Datenbankschicht zu verlagern. Auf diese Weise können Sie zum einen das Datenvolumen in der Applikationsschicht reduzieren und zum anderen signifikante Performancegewinne durch die Nutzung der neuen SAP-HANA-Datenbankfunktionen erzielen.

Entwicklung für SAP S/4HANA

Entwicklungen für SAP S/4HANA erfolgen künftig also sowohl auf der Ebene der Datenbank als auch weiterhin mithilfe der SAP-Programmiersprache ABAP in der Anwendungsschicht. Die wesentlichen Werkzeuge und ihre Eigenschaften stellen wir Ihnen in den folgenden Abschnitten vor. Der Fokus liegt dabei zunächst auf der Entwicklung der Geschäftslogik. Die neue Oberflächentechnologie SAP Fiori wird in Abschnitt 2.4 separat betrachtet.

2.2.1 SAP HANA Studio

Nachdem wir im vorigen Abschnitt gezeigt haben, wie rechenintensive Kalkulationen auf die Datenbankschicht verschoben werden, beschreiben wir in diesem Abschnitt, wie Komponenten entwickelt werden.

Während in den Zeiten vor der SAP-HANA-Datenbank ausschließlich über das SAP GUI (z. B. über die Transaktion SE80) entwickelt wurde, wird heute unter anderem das sogenannte *SAP HANA Studio* als Entwicklungsumgebung in der Datenbankschicht eingesetzt.

Eclipse als Entwicklungsumgebung

Das SAP HANA Studio basiert auf der Open-Source-Entwicklungsumgebung *Eclipse*. Diese Entwicklungsumgebung können Sie über das Hinzufügen von Plug-ins beliebig erweitern. Diese Erweiterbarkeit ist der Grund, warum SAP Eclipse als Basis für das SAP HANA Studio nutzt. Mit dem SAP HANA Studio können Sie datenbanknahe Entwicklungen wie SAP-HANA-Sichten und SQLScripts erstellen. Außerdem ermöglicht das SAP HANA Studio das Verwalten und Einrichten der Zugriffsberechtigungen.

Das SAP HANA Studio wird lokal auf dem Rechner des Anwenders installiert. Anschließend müssen Sie es über einen SAP-HANA-Datenbankbenutzer mit SAP HANA verbinden. Im Anschluss daran können Sie nahezu alle datenbanknahen SAP-Komponenten (z. B. Tabellen und Views) auch mithilfe des SAP HANA Studios verwenden.

Abbildung 2.6 zeigt einen Screenshot der Eclipse-Oberfläche des SAP HANA Studios. Dort können Sie Objekte sowohl grafisch als auch textuell modellieren. Das bedeutet, dass Sie Objekte z. B. in ABAP sowohl mithilfe der Oberfläche per Texteingabe (klassische Implementierung) als auch mithilfe eines grafischen Werkzeugs erstellen können.

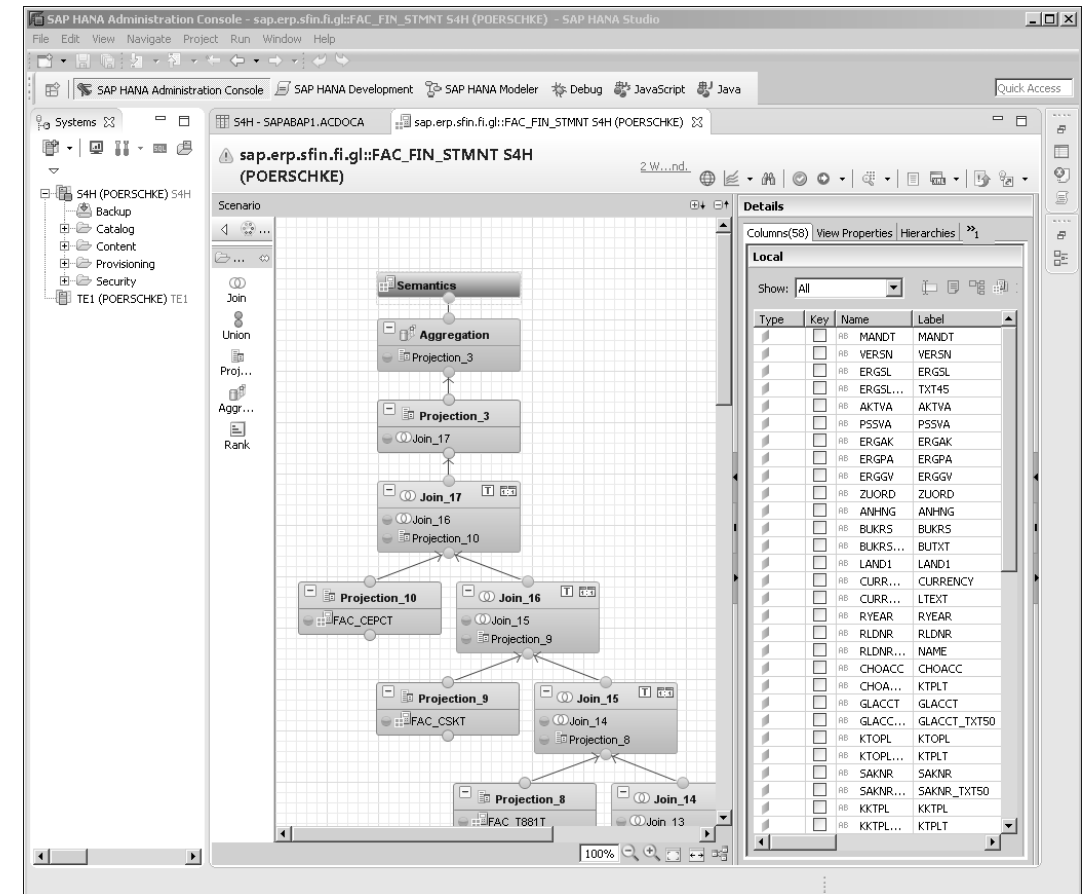


Abbildung 2.6 Das SAP HANA Studio

Wenn Sie dieses Werkzeug nutzen, können Sie Objekte erzeugen, ohne selbst Quelltext eingeben zu müssen. Den entsprechenden Quelltext erzeugt das Programm.

Die beiden wesentlichen Menüpunkte des SAP HANA Studios sind **Catalog** und **Content**. Diese beiden Verzeichnisse stellen wir Ihnen im Folgenden vor.

Unter **Catalog** verwalten und bearbeiten Sie alle SAP-HANA-Datenbankobjekte. Hier finden Sie alle physischen Datenstrukturen, Tabellen und Views wieder, die klassisch z. B. über SAP-Standardtransaktionen (z. B. Transaktion SE11 – Pflege der Datenbankobjekte und Transaktion SE16 – Anzeigen von Inhalten) verwaltet wurden. Neue Datenbanktabellen, Erweiterungen oder Views können so in Zukunft auch aus dem SAP HANA Studio heraus angelegt und bearbeitet werden. Die klassischen SAP-GUI-Oberflä-

Datenbankobjekte im Catalog

chen stehen Ihnen weiterhin zur Verfügung. Abbildung 2.7 zeigt ein Beispiel für eine Tabellendefinition (Tabelle ACDOCA) im SAP HANA Studio.

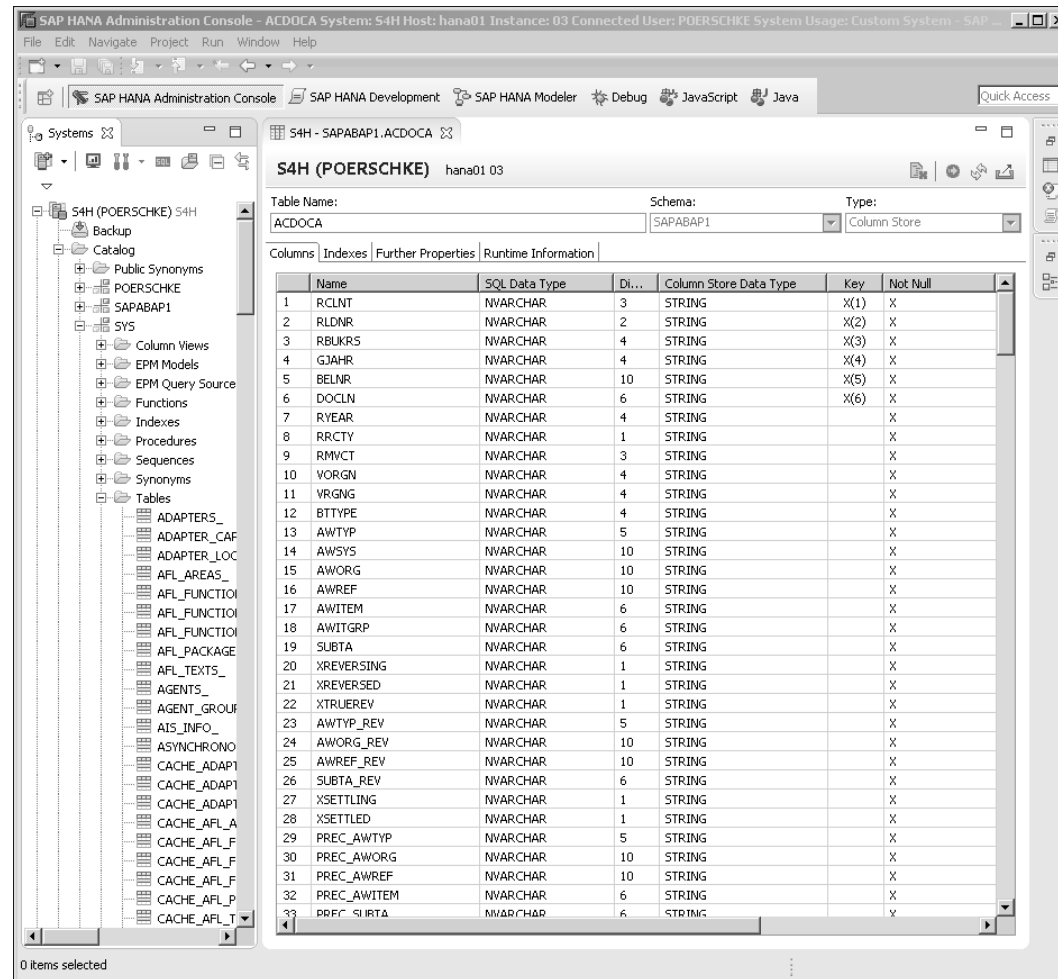


Abbildung 2.7 SAP HANA Studio – Catalog (Beispiel: Tabelle ACDOCA)

Datenbankmodelle im Content

Die wesentliche Neuerung unter SAP S/4HANA stellen allerdings die Funktionen des Contents dar: Unter **Content** ist es möglich, mit den SAP-HANA-Datenbankmodellen zu arbeiten. Diese SAP-HANA-Datenmodellierung stellt das Kernstück der Entwicklungen in der SAP-HANA-Datenbank dar. Das Ziel dieser Datenmodellierungen ist der Code Pushdown aus Logiken der Datenverarbeitung in Richtung der Datenbank (siehe Kapitel 3, »Prinzipien des Redesigns«).

2.2.2 Entwicklungskomponenten von SAP HANA

Das SAP HANA Studio stellt als Entwicklungskomponenten im Wesentlichen vier Werkzeuge zur Verfügung, die Sie im Folgenden kennenlernen:

- SQL-Prozeduren (Procedures)
- Entscheidungstabellen (Decision Tables)
- Analyseberechtigungen (Analytic Privileges)
- SAP-HANA-Sichten (Views)

Diese Objekte dienen der datenbanknahen Datenverarbeitung und können ausschließlich über das SAP HANA Studio verwaltet werden.

SQL-Prozeduren

Mit der Skriptsprache *SQLScript* stellt SAP HANA ein Werkzeug zur Implementierung von Datenbankprozeduren und Berechnungssichten zur Verfügung. Diese Skriptsprache ermöglicht es, modulare Datenbankfunktionen zu erstellen und über eine imperative Sprache zu implementieren.

Die so definierten SQL-Programme können als wiederverwendbare Prozeduren gekapselt und mit Input- und Output-Parametern versehen werden. Im Anschluss können sie bei der Definition von SAP-HANA-Views genutzt und in SAP-S/4HANA-Programmen angesteuert werden.

So ist es möglich, komplexe Analysefunktionen zu modularisieren und wiederzuverwenden. Abbildung 2.8 stellt die Entwicklung einer SQL-Prozedur dar. Im Hauptteil findet die eigentliche Implementierung der Prozedur statt, und auf der rechten Seite werden die Input- und Output-Parameter definiert.

Bei der Nutzung von *SQLScript* kann der Entwickler Standardfunktionen aus verschiedenen Programmbibliotheken nutzen. Ein Beispiel hierzu stellt die Business Function Library (BFL) dar.

Die BFL ist eine Sammlung von Funktionen aus der Finanzwirtschaft. Mit diesen Funktionen lassen sich komplexe und performancekritische Algorithmen direkt in der Datenbank ausführen. Das kann zum einen deutliche Verbesserungen in der Laufzeit von Anwendungsprogrammen bringen, zum anderen kann die Verwendung der Funktionen zu einer Vereinfachung und Beschleunigung bei der Anwendungsentwicklung führen.

Werkzeuge des SAP HANA Studios

2

SQLScript

Nutzung von Programm-bibliotheken

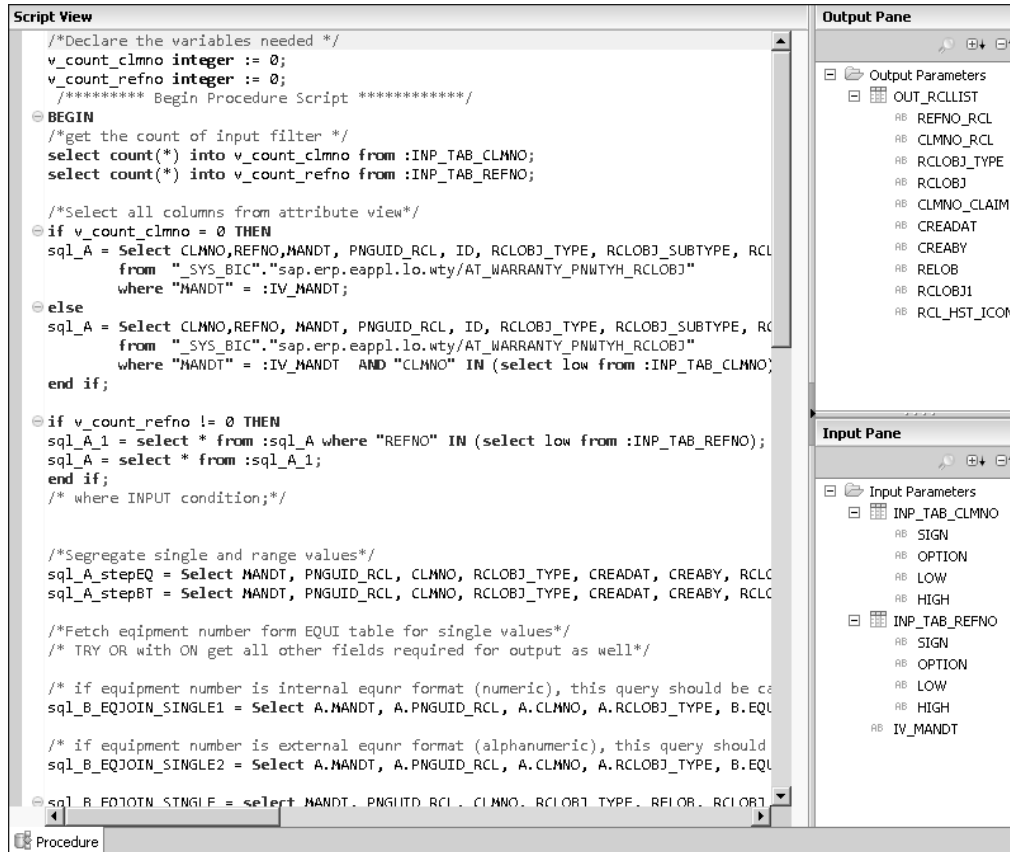


Abbildung 2.8 SQL-Prozeduren

Entscheidungstabellen

SAP-HANA-Entscheidungstabellen ermöglichen es Ihnen, Teile Ihrer Programmlogik in flexibel definierbare Regeln auszugliedern. Die Verwendung von Entscheidungstabellen hat folgende Vorteile:

- Die Komplexität von Anwendungen wird durch klar lesbare und verständliche Tabellen verringert.
- Es lassen sich flexible Strukturen generieren, die es ermöglichen, auf neue Sachverhalte zu reagieren, ohne die Programmierung einer Anwendung anpassen zu müssen.

Über Entscheidungstabellen definieren Sie, basierend auf einem oder mehreren Parametern, welches Ergebnis aus diesen Tabellen automatisiert abgeleitet werden soll. Dazu können feste Wertzuordnungen oder auch flexible Formeln (mithilfe des integrierten Formeleditors) zum Einsatz kommen.

Vorteile von Entscheidungstabellen

Auf diese Weise benötigen Sie keine Programmierkenntnisse, um Entscheidungstabellen zu verstehen und zu erweitern. Abbildung 2.9 zeigt die Definition einer Entscheidungstabelle.

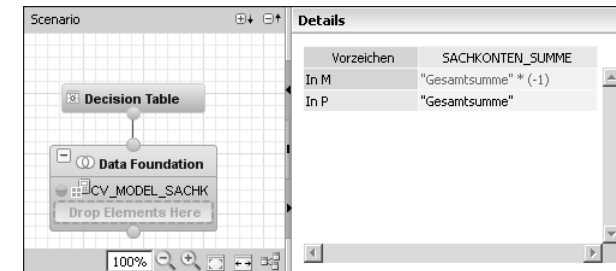


Abbildung 2.9 Beispiel für eine Entscheidungstabelle

Analyseberechtigungen

Analyseberechtigungen erlauben es, den Zugriff auf Daten von SAP-HANA-Sichten einzuschränken. So können zu bestehenden Attribut-, Analyse- oder Berechnungssichten anhand ihrer Ausprägungen flexibel Zugriffsberechtigungen definiert werden. Diese Berechtigungen schränken (gegebenenfalls zeitabhängig) den Zugriff auf die Daten ein. So definierte Analyseberechtigungen können im Anschluss beliebigen SAP-HANA-Anwendern zugewiesen werden.

Diese Analyseberechtigungen können nicht auf SAP-Datenbanktabellen angewendet werden. Abbildung 2.10 zeigt die Definition einer Analyseberechtigung. Hier definiert man den Zeitraum, für den eine Berechtigung erteilt wird, sowie die Datenräume, für die eine Berechtigung erteilt oder eingeschränkt wird.

Flexible Zugriffsberechtigungen

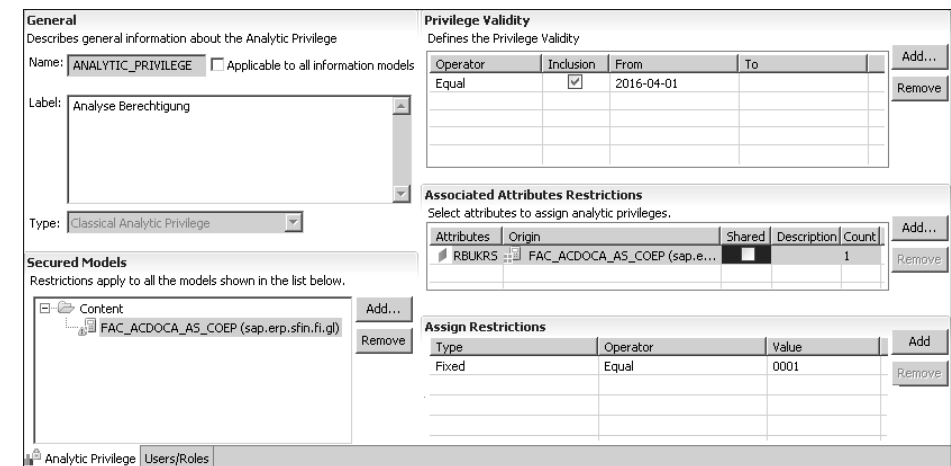


Abbildung 2.10 Analyseberechtigungen

Attribut-, Analyse- und Berechnungssicht

Sichten (Views)

Es ist möglich, drei Arten von Datenbanksichten zu definieren. Diese Sichten können im Anschluss z. B. in Analysewerkzeugen wie SAP Lumira und anderen Reporting-Werkzeugen des SAP-BusinessObjects-Portfolios verwendet werden (siehe Abschnitt 2.3, »Analysewerkzeuge von SAP HANA«):

■ **Attributsicht**

Eine Attributsicht ist eine wiederverwendbare Sicht auf Stammdaten. Hierbei können Joins über mehrere Tabellen definiert werden. Diese Joins können in Analyse- und Berechnungssichten wiederverwendet werden. In der Regel werden Attributsichten genutzt, um die Stammdaten mit den zugehörigen Texttabellen zu verknüpfen. Es ist aber ebenso möglich, verschiedene Stammdatentabellen miteinander zu verknüpfen.

Abbildung 2.11 zeigt die Definition einer Attributsicht, bei der zwei Tabellen (SKA1 und SKAT) miteinander verbunden werden.

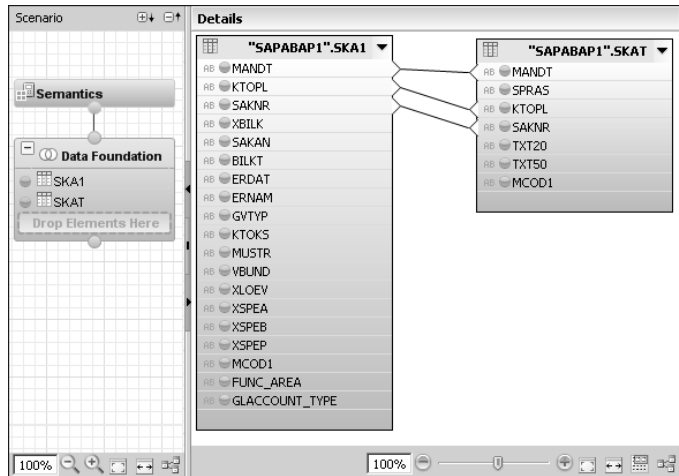


Abbildung 2.11 Attributsicht (Stammdaten zum Sachkonto)

■ **Analysesicht**

Analysesichten stellen eine Kombination aus Merkmalen und Kennzahlen dar. Sie sind mit OLAP-Cubes in SAP BW vergleichbar. Eine Analysesicht besteht in der Regel aus einer Kennzahlentabelle (Faktentabelle), die um ergänzende Daten (Dimensionstabellen) erweitert wird.

Ein Beispiel für eine derartige grafische Modellierung einer Analysesicht ist in Abbildung 2.12 dargestellt.

■ **Berechnungssicht**

In Berechnungssichten ist es möglich, über Entscheidungstabellen, Aggregationen, Berechnungsregeln und SQLScript-Funktionen flexible Datenberechnungen durchzuführen und bereitzustellen.

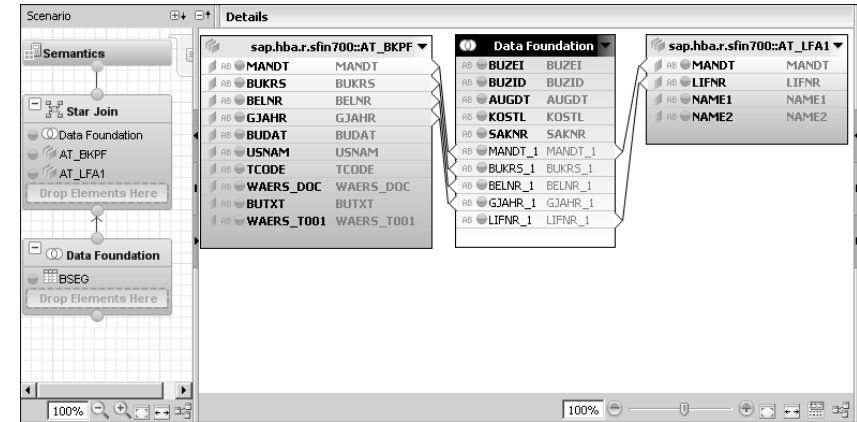


Abbildung 2.12 Analysesicht (Einzelposten in FI)

Im Rahmen der Berechnungssichten können Sie entweder verschiedene andere Sichten und Tabellen über eine grafische Modellierung miteinander in Beziehung setzen und durch Standardfunktionen bearbeiten, oder Sie implementieren alle Berechnungen über ein SQLScript.

Abbildung 2.13 zeigt ein Beispiel für die Definition einer Berechnungssicht.

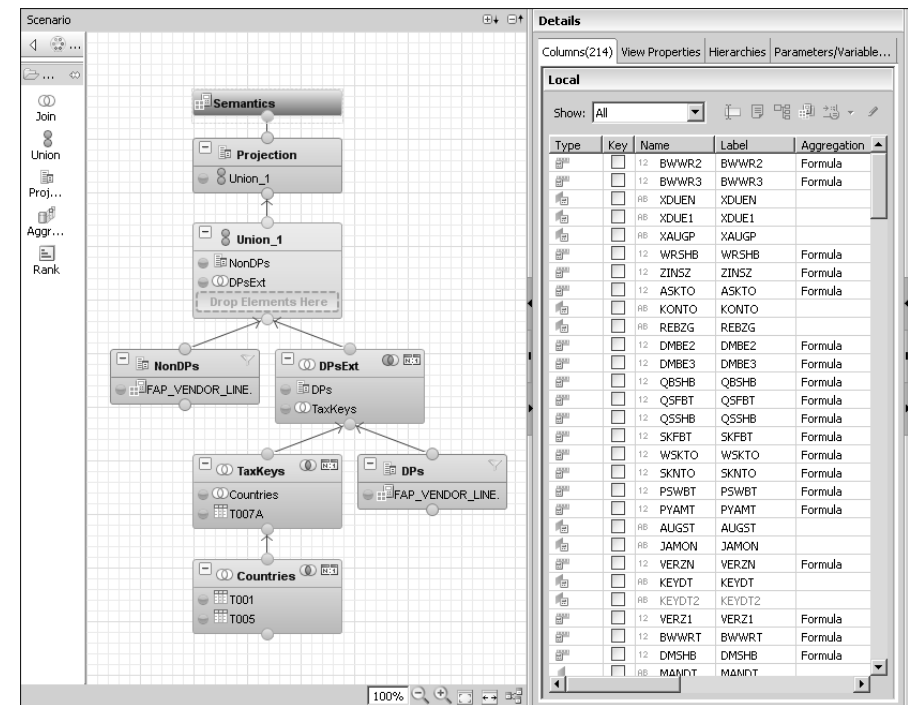


Abbildung 2.13 Berechnungssicht (SAP_VENDOR_LINE_ITEMS)

2.2.3 ABAP-Entwicklung

Unabhängig davon, ob Sie SAP S/4HANA in der Cloud benutzen oder ob Sie es als On-Premise-Lösung einsetzen: Das Backend eines SAP-S/4HANA-Systems ist weiterhin in ABAP entwickelt. Befürchtungen, dass in Zukunft Ihr ABAP-Know-how obsolet werden könnte, sind also unbegründet. Es ist jedoch erforderlich, dass Sie sich bezüglich der ABAP-Entwicklung mit einigen Neuerungen auseinandersetzen, damit Sie die Potenziale von SAP S/4HANA optimal nutzen können. Einige dieser Neuerungen stellen wir in den folgenden Abschnitten vor.

Neue ABAP-Funktionen für den Code Pushdown

Die In-Memory-Technik von SAP HANA allein garantiert noch keine Verbesserung der Laufzeit. Die Laufzeit von Datenbankabfragen wird nicht nur durch die Performance der eigentlichen Suche auf der Datenbank beeinflusst. Ein weiterer entscheidender Faktor bei der Laufzeit ist die Datenmenge, die von der Datenbank zum Applikationsserver übertragen wird. Indem die übertragene Datenmenge möglichst klein gehalten wird, kann die Laufzeit deutlich reduziert werden. Deshalb wurde die Programmiersprache ABAP deutlich erweitert und hinsichtlich des Code Pushdowns optimiert.

Nutzung von Open SQL

Eine der wichtigsten Neuerungen ist eine umfangreiche Erweiterung von *Open SQL*. SQL ist eine Datenbanksprache zum Bearbeiten und Abfragen von Daten eines Datenbankmanagementsystems. SQL ist zwar standardisiert, trotzdem gibt es je nach Datenbankmanagementsystem unterschiedliche SQL-Dialekte. Open SQL ist ein proprietäres SQL-Derivat von SAP, das unabhängig von der zugrunde liegenden Datenbank ist und von der Laufzeitumgebung in den entsprechenden SQL-Dialekt, das sogenannte *Native SQL*, übersetzt wird. Im Zuge der Erweiterung von Open SQL wurde eine veränderte Syntax eingeführt. Aus Gründen der Abwärtskompatibilität ist die alte Syntax zwar weiterhin gültig, neue Sprachelemente von Open SQL lassen sich mit der alten Syntax aber nicht nutzen.

Neue Möglichkeiten mit der Open-SQL-Erweiterung

Als Beispiele für neue Möglichkeiten, die sich Ihnen durch die Open-SQL-Erweiterung eröffnen, möchten wir Ihnen folgende nennen:

■ Inline-Deklaration

Um in einem ABAP-Programm mit Daten aus der Datenbank arbeiten zu können, müssen diese Daten zunächst über Select-Anweisungen von der Datenbank gelesen und in sogenannten Arbeitsbereichen abgelegt werden. Ein Arbeitsbereich ist dabei eine Variable des ABAP-Programms, die die Struktur der Felder der Datenbankabfrage abbildet. Bisher konnten

der Arbeitsbereich und die Datenbankabfrage nur getrennt voneinander definiert werden. Das Problem dabei war, dass bei einer Erweiterung der Datenbankabfrage um weitere Felder auch immer die Definition des Arbeitsbereichs angepasst werden musste. Diese Anforderung führte dazu, dass der Einfachheit halber oft alle Felder einer Datenbanktabelle gelesen wurden, auch wenn nur ein Teil der Felder wirklich im Programm benötigt wurde. Über die Inline-Deklaration ist jetzt die Definition des Arbeitsbereichs direkt mit der Definition der Datenbankabfrage verknüpft. Eine Änderung der Datenbankabfrage ändert also automatisch auch die Struktur des Arbeitsbereichs. Dies erleichtert es dem ABAP-Entwickler, nur die Spalten einer Datenbanktabelle zu lesen, die in der weiteren Verarbeitung auch wirklich verwendet werden.

■ Arithmetische Ausdrücke

In manchen Anwendungsfällen ist die Weiterverarbeitung eines Datensatzes von einer Berechnung über mehrere Felder einer Datenbanktabelle abhängig. Bisher konnte eine Geschäftslogik dieser Form in einem ABAP-Programm nur so umgesetzt werden, dass alle Daten gelesen werden und dann die Berechnung im ABAP-Programm durchgeführt wird. Das führte dazu, dass unter Umständen mehr Daten gelesen werden mussten, als für die weitere Verarbeitung nötig waren. Über arithmetische Ausdrücke ist es nun möglich, eine Reihe an Berechnungen schon in der Datenbankabfrage zu prozessieren, was wiederum zu einer Reduzierung der zu selektierenden Datenmenge führen kann.

■ Fallunterscheidungen

Mit der Open-SQL-Erweiterung können auch Fallunterscheidungen in einer Datenbankabfrage realisiert werden. Zum Beispiel ist es damit möglich, abhängig von der Höhe eines Betrags unterschiedliche Koeffizienten für Berechnungen heranzuziehen und damit Geschäftslogik bereits in der Datenbankabfrage abzubilden.

Eine weitere Neuerung für ABAP-Entwickler sind die sogenannten *ABAP Core Data Services* oder auch kurz *ABAP CDS* genannt. ABAP CDS sind eine weitestgehend datenbankunabhängige Implementierung der allgemeinen CDS, die wir Ihnen bereits in Verbindung mit der Entwicklung auf der SAP-HANA-Datenbank vorgestellt haben. Zur Abgrenzung werden die allgemeinen CDS vielfach auch als HANA CDS bezeichnet.

Core Data Services und CDS Views

ABAP CDS sind voll in die neue auf Eclipse basierende ABAP-Entwicklungsumgebung ABAP Development Tools (ADT) integriert. Der Vorteil dieser Einbettung liegt darin, dass die Entwicklung einer CDS-Entität und deren Verwendung in einem ABAP-Programm in derselben Entwicklungsumgebung umgesetzt werden und beides gemeinsam in ein übergelagertes Sys-

tem transportiert werden kann. In der ABAP-Entwicklung werden die Core Data Services unter anderem dafür genutzt, um sogenannte *CDS Views* anzulegen, deren Funktionalität weit über die Fähigkeiten der aus Transaktion SE11 bekannten Views hinausgeht. Dies ist eine weitere Möglichkeit für den Entwickler, um Programmlogik in die Datenbankschicht zu verlagern.

Die Funktionen von Open SQL mit den Erweiterungen und die CDS Views sind bis auf wenige Ausnahmen als gleichwertig zu betrachten. Wenn eine Datenbankoperation wiederverwendbar sein soll, wird man sich meist für CDS Views entscheiden. Wird die entsprechende Datenbankoperation nur einmal benötigt, ist Open SQL in den meisten Fällen die bevorzugte Wahl.

ABAP Managed Database Procedures

Insbesondere im Hinblick auf den Code Pushdown sind einige Erweiterungen an Open SQL vorgenommen worden. Diese ermöglichen den Entwicklern, rechenintensive Vorgänge in die Datenbankschicht zu verlagern. ABAP- und SAP-HANA-Entwicklungen wachsen auf diese Weise deutlich näher zusammen. Als Beispiel sind an dieser Stelle auch die sogenannten *ABAP Managed Database Procedures (AMDP)* zu nennen. Mit AMDP lassen sich Datenbankprozeduren, wie wir sie Ihnen bereits in Abschnitt 2.2.2 über die Entwicklung auf der SAP-HANA-Datenbank vorgestellt haben, in der ABAP-Entwicklungsumgebung anlegen und pflegen. Technisch gesehen handelt es sich bei AMDP um SQLScript-Coding, das in eine ABAP-Klasse gekapselt wird. Transportiert werden AMDP zusammen mit ABAP-Programmen über die bekannten Transportwege. Nach dem Transport wird das eingebettete SQLScript-Programm einer AMDP an die entsprechende Datenbankinstanz deployt.

ABAP-Entwicklungsumgebung

ABAP Development Tools

Wie bereits beim SAP HANA Studio für die Entwicklung auf Datenbankebene legt SAP zukünftig auch bei der Entwicklung der Anwendungsschicht die Open-Source-Plattform Eclipse zugrunde. Auch dafür werden Plug-ins – die sogenannten *ABAP Development Tools (ADT)* – angeboten. Diese Werkzeuge verwandeln die Eclipse-Basis in eine ABAP-Entwicklungsumgebung. Abbildung 2.14 zeigt die ABAP-Entwicklung innerhalb der Eclipse-Umgebung. Auf der linken Seite wird dabei die Struktur der Entwicklungspakete und Objekte dargestellt, und im Hauptfenster findet die eigentliche Programmierung statt.

ABAP Workbench vs. ABAP in Eclipse

Die bekannte *ABAP Workbench* können Sie zwar weiterhin nutzen, neue Funktionen (wie z. B. die Implementierung von CDS Views oder AMDP) werden aber von der ABAP Workbench nicht mehr unterstützt und können nur über ABAP in Eclipse, wie die ABAP Development Tools auch gerne genannt werden, angelegt werden.

Die ABAP Development Tools werden mehr und mehr ausgebaut, damit auch alle Entwicklungsobjekte in Eclipse angelegt und gepflegt werden können. Für Objekte, die noch nicht direkt in Eclipse angelegt werden können, wird eine eingebettete SAP-GUI-Session mit der entsprechenden Transaktion in Eclipse geöffnet, sodass Sie dafür nicht in eine andere Entwicklungsumgebung wechseln müssen.

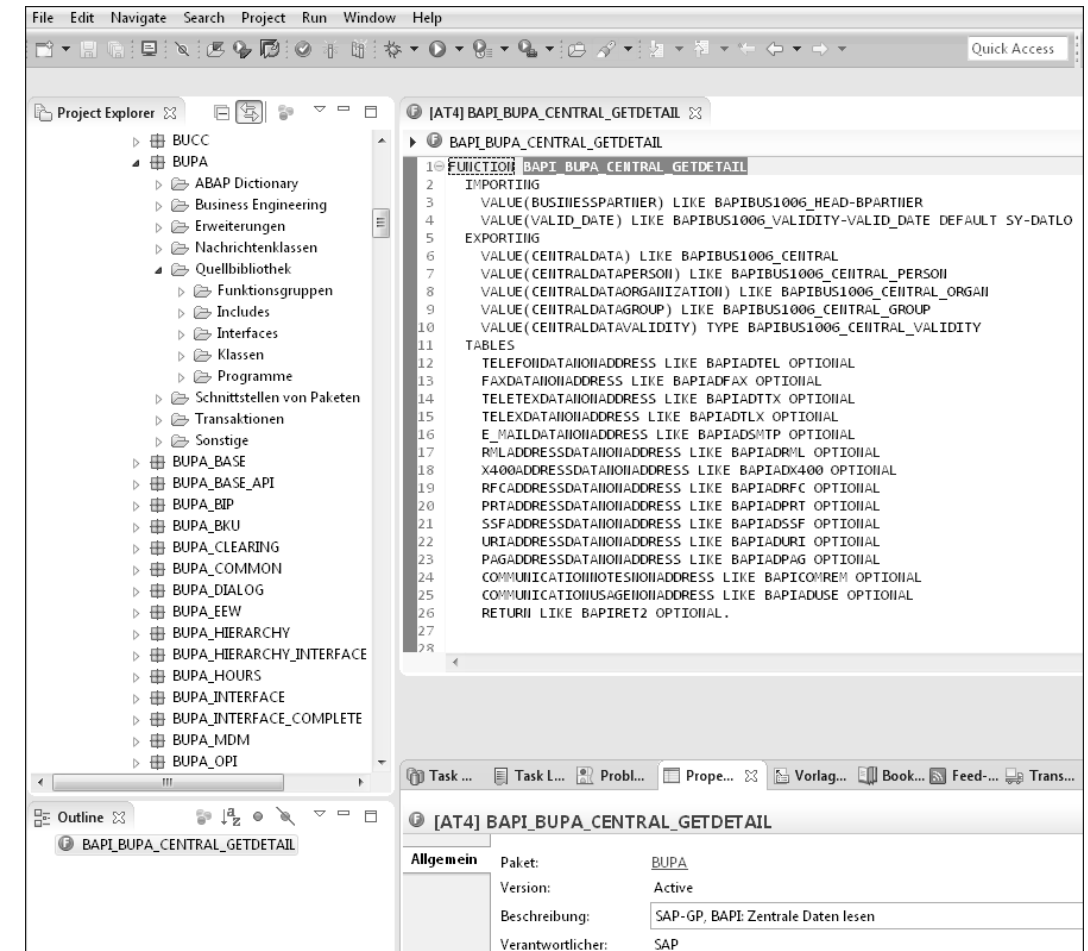


Abbildung 2.14 ABAP-Entwicklung in Eclipse

In Abbildung 2.15 sehen Sie, wie eine SAP-GUI-Transaktion eingebettet wird. Sie zeigt ein Data-Dictionary-Objekt. Durch die Einbettung von SAP-GUI-Transaktionen wird sichergestellt, dass ein Entwickler alle notwendigen Tätigkeiten direkt in Eclipse durchführen kann, ohne permanent zwischen zwei unterschiedlichen Umgebungen wechseln zu müssen.

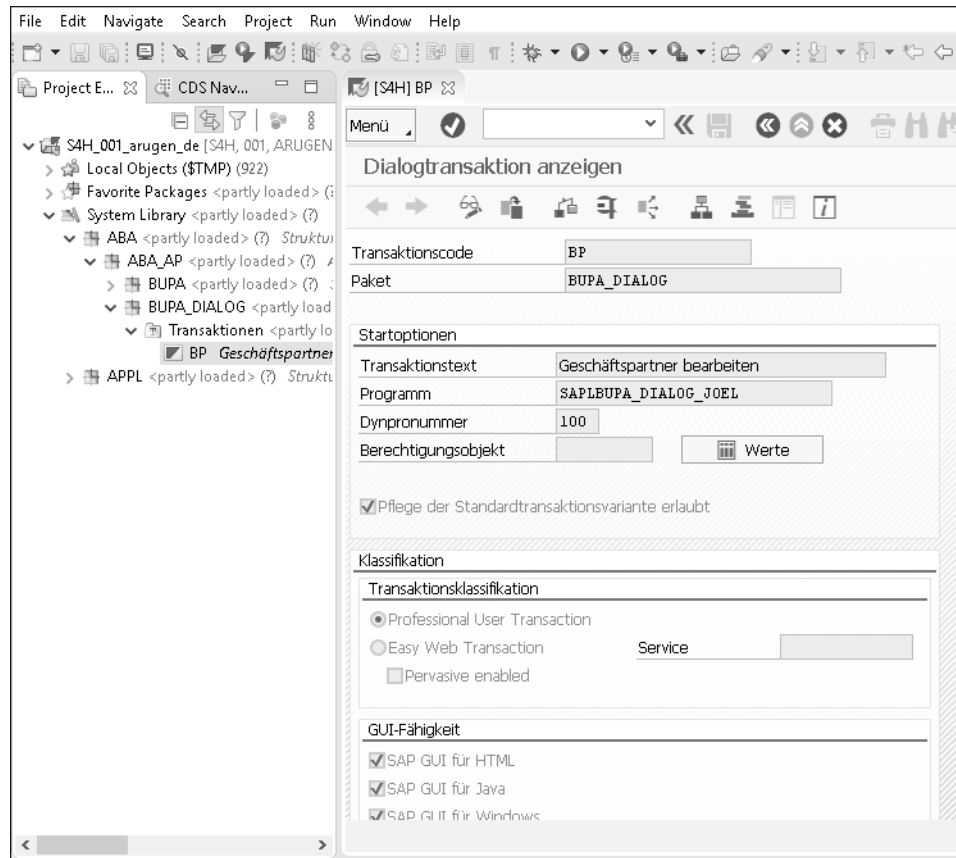


Abbildung 2.15 Eingebettetes SAP GUI in Eclipse

2.2.4 Erweiterbarkeit von SAP S/4HANA

In einem Unternehmen besteht immer wieder die Notwendigkeit, die Geschäftsprozesse zu optimieren, um wettbewerbsfähig zu bleiben. In solchen Fällen müssen Sie auch Ihre Unternehmenssoftware den neuen Bedürfnissen anpassen. Dies ist in vielen Fällen nicht ohne Kundenerweiterungen, also kundeneigene Programmierung, möglich. Ein Grund für die große Akzeptanz von SAP-Lösungen in den Unternehmen waren daher auch immer schon die vielfältigen Möglichkeiten, kundenspezifische Erweiterungen vorzunehmen oder Lösungen von Drittanbietern (sogenannte Add-ons) in das SAP-ERP-System zu integrieren.

Erweiterungsmöglichkeiten

Die Möglichkeiten, die Sie für die Erweiterung von SAP S/4HANA haben, unterscheiden sich, je nachdem, welche Deploymentform Sie nutzen.

In der On-Premise-Version von SAP S/4HANA entscheiden Sie selbst, ob und wann ein Releasewechsel oder System-Update durchgeführt wird. Bei Problemen und Verzögerungen beim Releasewechsel, die durch eine zu starke Abhängigkeit zwischen SAP-Kernsystem und Kundenerweiterung hervorgerufen wurden, sind nur Sie betroffen. Deshalb haben Sie hier auch deutlich mehr Freiheiten, was die Erweiterung Ihres SAP-Systems angeht. Sie können über das Erweiterungs-Framework alle von SAP ERP bekannten Möglichkeiten wie User Exits, Events und Customer Enhancements nutzen und auch Modifikationen am Standardcode vornehmen. Diese Erweiterungsmöglichkeiten stehen Ihnen auch dann zur Verfügung, wenn Sie den Service SAP HANA Enterprise Cloud nutzen.

Wenn Sie SAP S/4HANA Cloud Extended in der Cloud nutzen, sind Sie gegenüber den beiden zuvor vorgestellten Optionen in zwei Dimensionen eingeschränkt: Modifikationen können Sie hier nicht vornehmen. Für implizite Enhancements brauchen Sie zwar keinen Modifikationsschlüssel, sie sind an dieser Stelle aber wie Modifikationen zu betrachten und deshalb nicht erlaubt. Mit impliziten Enhancements haben Sie die Möglichkeit, am Anfang und am Ende eines jeden SAP-Programmierblocks Erweiterungen einzubauen. Da bei einem Upgrade von SAP nicht garantiert werden kann, dass der Programmierblock und damit der Erweiterungspunkt weiterhin existiert und der Zugriff auf dieselben Daten zur Verfügung steht, müssen implizite Enhancements nach einem Upgrade wie Modifikationen behandelt werden. Sie müssen für jedes implizite Enhancement in Transaktion SPAU_ENH einen Abgleich durchführen. Abgesehen davon können Sie das Erweiterungs-Framework vollumfänglich nutzen – hierbei wirkt auf Sie nur einschränkend, dass Sie zu den verabredeten Zeitpunkten für Releasewechsel die Funktionsfähigkeit Ihrer Erweiterungen überprüfen müssen.

Am stärksten schränkt Sie die Lösung SAP S/4HANA Cloud Essentials hinsichtlich der Erweiterungen ein. Sie stellt ein vollständiges ERP-System in der Cloud dar. SAP betreut den Betrieb und die Wartung des Systems. Releasewechsel und System-Updates werden von SAP in regelmäßigen Abständen durchgeführt und dürfen durch Kundenanpassungen nicht behindert werden. Deshalb ist hier eine klare Trennung von SAP-Kernsystem und Kundenentwicklung zwingend erforderlich. Die Stabilität bezüglich Releasewechseln hat in der Cloud eine deutlich höhere Bedeutung und stellt einen limitierenden Faktor bei der Erweiterbarkeit dar.

Erweiterungen in der Cloud-Version Um diesen Herausforderungen bezüglich der Erweiterungen der SAP S/4HANA Cloud Essentials Rechnung zu tragen, hat SAP ein neues Konzept für Kundenerweiterungen für SAP S/4HANA entwickelt, das sich an den folgenden Anforderungen orientiert:

- Anwender, Anwendungsbetreuer und Entwickler müssen Erweiterungen in ihrem Verantwortungsbereich ohne Risiko vornehmen können.
- Es muss eine strikte Trennung zwischen Kundenerweiterung und Kernsystem bestehen, sodass weder ein Releasewechsel durch kundenspezifische Erweiterungen blockiert noch die Funktionalität einer Kundenerweiterung nach dem Releasewechsel beeinträchtigt wird.
- Es müssen Werkzeuge angeboten werden, die sowohl die Anforderungen des Kunden bezüglich Flexibilität als auch die Interessen des Dienstbieters bezüglich Stabilität berücksichtigen.

Typen von Erweiterungen Das Ergebnis ist ein neues Erweiterungskonzept von SAP für SAP S/4HANA. Dieses Konzept unterscheidet zwischen den beiden folgenden Typen von Erweiterungen:

- **In-App-Erweiterungen**
Erweiterungen innerhalb des Systems. Die Erweiterungen innerhalb des Systems werden noch einmal in zwei Untergruppen unterteilt:
 - *Key-User-Erweiterungen*: Mit Key-User-Erweiterungen sind Erweiterbarkeitsfunktionen gemeint, die von Anwendungsexperten und Entwicklern über Erweiterbarkeits-Apps realisiert werden können.
 - *Klassische Erweiterungen*: Klassische Erweiterungen sind ein Sammelbegriff für alle bereits aus anderen SAP Business Suites bekannten Erweiterungstechniken wie BTE, BDT, BAdI und Enhancements. Für die Umsetzung dieser Erweiterungen ist ein hohes Maß an technischem Know-how notwendig. Deshalb sind die klassischen Erweiterungen eher Entwicklern vorbehalten.
- **Side-by-Side-Erweiterungen**
Erweiterungen von außen, die lose über öffentliche Schnittstellen mit dem System verbunden sind

Abbildung 2.16 zeigt den grundsätzlichen Unterschied zwischen den In-App-Erweiterungen und einer Side-by-Side-Erweiterung. Im Rahmen der Key-User-Erweiterung bietet SAP mehrere Erweiterbarkeits-Apps an, um kundeneigene Anpassungen vornehmen, testen und ins Produktivsystem importieren zu können.

	on premise	Private Cloud	Cloud Extended	Cloud Essentials
In-App Extensibility				
Key User Extensibility • Erweiterung der Benutzeroberfläche, Kundenfelder und -tabellen, Berichte und Formulare • Anpassung der Geschäftslogik	✓	✓	✓	✓
Classic Extensibility • voller Zugriff auf ABAP und SAP HANA • Erweiterungen wie in der SAP Business Suite	✓	✓	✓	✗
Modifikationen • implizite Enhancements • Modifikationen von SAP-Programmen	✓	✓	✗	✗
Side-by-Side Extensibility				
• SAP Cloud Platform als Erweiterungsplattform • Integration von Cloud Lösungen von SAP oder Drittanbietern • Zugriff auf SAP S/4 HANA nur über öffentliche Schnittstellen	✓	✓	✓	✓

Abbildung 2.16 In-App-Erweiterung und Side-by-Side-Erweiterung

Key-User-Erweiterung

Die im Erweiterungskonzept vorgesehene Key-User-Erweiterung ist in SAP S/4HANA in Form von sogenannten Erweiterbarkeits-Apps umgesetzt worden. Mit diesen Apps können unter anderem die folgenden Erweiterungen vorgenommen werden:

Benutzerdefinierte Felder und Logik

- Sie können benutzerdefinierte Felder, Erweiterungen von Datenquellen und Entwicklungen kundeneigener Geschäftslogik anlegen und damit erweiterbare Anwendungen erweitern.
- Sie können über benutzerdefinierte CDS Views die Daten verschiedener Datenquellen zusammenführen und die Tabellenfelder nach Ihren Wünschen anordnen. Mittels der CDS Views können Sie wiederum eigene analytische Abfragen definieren.
- Sie können eigene Business-Objekte mit dazugehörigen UIs anlegen und damit eigene Anwendungen realisieren.

Nicht jede Anwendung lässt sich über die Erweiterbarkeits-Apps anpassen. Damit eine Erweiterung einer Anwendung möglich ist, muss diese Anwendung als erweiterbar gekennzeichnet sein. Informationen über die Erwei-

terbarkeit einer App können Sie dem entsprechenden Eintrag in der SAP Fiori Apps Reference Library entnehmen.

Sind Sie Kunde einer SAP-S/4HANA-Cloud-Lösung, dann können Sie als Einstieg für die Erweiterung das *Extensibility Cockpit* nutzen. In dieser neuen App können Sie nach Erweiterungsoptionen suchen und direkt in die entsprechende Erweiterbarkeits-App abspringen.

Eine weitere Voraussetzung für die Key-User-Erweiterung ist, dass diese in Ihrem System aktiviert wurde. Standardmäßig ist die Erweiterbarkeit für Key-User deaktiviert. Wie man die Erweiterbarkeit aktivieren kann, hängt von der Art Ihres Systems ab. Weitere Informationen können Sie SAP-Hinweis 2283716 entnehmen.

Benutzerdefinierte Felder und Logik

Von den Erweiterbarkeits-Apps ist besonders die App *Benutzerdefinierte Felder und Logik* hervorzuheben. Diese App bietet drei verschiedene Erweiterungsoptionen. Zum einen können mit dieser App kundeneigene Felder definiert werden. Nach dem Anlegen und Aktivieren der Felder können diese in Benutzeroberflächen, Reports, E-Mail-Vorlagen und Formularvorlagen verwendet werden. Zum anderen können Sie die Datenquelle einer Anwendung um weitere Standardfelder erweitern. Dabei müssen Sie selbst darauf achten, dass der Endbenutzer durch die Erweiterung der Datenquelle weiterhin nur die Daten sehen darf, für die er berechtigt ist. Die dritte Erweiterungsoption ist das Anlegen benutzerdefinierter Geschäftslogik, die für die Erweiterung einer Anwendung verwendet werden kann. Für die Implementierung benutzerdefinierter Logik steht Ihnen ein Webeditor zur Verfügung, der Ihnen den bekannten Komfort hinsichtlich Autovervollständigung, Syntaxprüfung und Formatierung (Pretty Printer) bietet. Als Programmiersprache ist eine eingeschränkte ABAP-Version, das sogenannte *ABAP für Key-User*, implementiert. Eingeschränkt ist hier vor allem der Zugriff auf das SAP-Kernsystem. Nur von SAP freigegebene Programmteile können aufgerufen werden. Direkte Datenänderungen über OpenSQL-Befehle sind nicht möglich. Eine Änderung von Anwendungsdaten kann damit nur über von SAP dafür vorgesehene Schnittstellen erfolgen. Ein direktes Lesen ist nur für freigegebene Datenbanktabellen zulässig. Damit ist sichergestellt, dass benutzerdefinierte Logik, die in ABAP für Key-User eingebaut wurde, nur lose mit dem SAP-Kernsystem verknüpft ist.

Bei der On-Premise-Version erfolgt der Transport der Erweiterungen über das normale Transportwesen. Dazu müssen die Erweiterungen über eine weitere Erweiterungs-App einem Transportauftrag zugeordnet werden. Bei einem Cloud-Essentials-System gibt es wiederum je eine App für den Export bzw. Import Ihrer Erweiterungen.

Klassische Erweiterungen

Mit *klassischen Erweiterungen* sind die Erweiterungen gemeint, die Sie bereits aus der SAP Business Suite kennen. Bei den klassischen Erweiterungen haben Sie den vollen Zugriff auf ABAP und die SAP-HANA-Datenbank. Diese Form der Erweiterung ist in der SAP-S/4HANA-Cloud-Essentials-Version nicht möglich. Entscheiden Sie sich insbesondere für eine On-Premise-Installation von SAP S/4HANA, haben Sie also weiterhin alle bekannten Möglichkeiten, das System zu erweitern. Kundenerweiterungen aus einer älteren SAP Business Suite können übernommen werden.

In SAP S/4HANA wurde jedoch das Datenmodell so weit verändert, dass Ihnen einige Datenbanktabellen nicht mehr zur Verfügung stehen. Sie müssen Ihre Kundenerweiterungen also möglicherweise vor einer Migration dem vereinfachten Datenmodell von SAP S/4HANA anpassen. Was Sie für die Migration von Kundenerweiterungen tun müssen und wie weit SAP Sie dabei unterstützt, werden wir Ihnen in Kapitel 6, »Umstellungsszenarien und prototypischer Ablauf einer Migration«, näher erläutern.

Auch wenn Sie als On-Premise-Kunde bezüglich Ihrer Entwicklungen so weiterverfahren können, wie Sie es von der SAP Business Suite her gewohnt sind, sollten bei Ihren Kundenerweiterungen künftig die in der Cloud möglichen Erweiterungstechniken erste Option sein. Das hat folgende Gründe: Zum einen verringert sich bei einer strikten Trennung von SAP-Kernsystem und Kundenerweiterungen der Test- und Anpassungsaufwand, der vor einem Releasewechsel notwendig wird. Dadurch können Sie in Zukunft schneller Neuerungen von SAP in Ihrem System einspielen und nutzen. Zum anderen können Sie die aufwendige Umwandlung von klassischer Erweiterung in cloudkompatible Erweiterungsoptionen vermeiden, sollten Sie für die Zukunft den Weg in die Cloud planen.

Side-by-Side-Erweiterung

Bei der Side-by-Side-Erweiterung nutzen Sie die SAP Cloud Platform als Erweiterungsplattform (siehe Abschnitt 2.5), um externe Anwendungen zu entwickeln. Die Kommunikation mit dem Kernsystem erfolgt über öffentliche Standardschnittstellen, sogenannte APIs (Application Programming Interface), oder über kundeneigene Schnittstellen, basierend auf benutzerdefinierten Business-Objekten. Mit dieser Erweiterungsplattform haben Sie als Entwickler volle Flexibilität im Umfeld von Java, HTML5 und nativen SAP-HANA-Services. Mit der neu von SAP eingeführten *SAP Cloud Platform ABAP Environment* ist jetzt auch eine ABAP-Entwicklung in der Cloud möglich. Sie können z. B. neue Anwendungen auf dieser Basis entwickeln und auf der SAP Cloud Platform als Laufzeitumgebung betreiben.

Anpassung von Erweiterungen für SAP S/4HANA

SAP Cloud Platform

Neben individuellen Anwendungen des Kunden können aber auch Cloud-Lösungen von SAP (z. B. SAP SuccessFactors oder SAP Ariba) oder von Drittanbietern integriert werden. Mit dem SAP-Kernsystem sind diese Erweiterungen lose über öffentliche Schnittstellen verknüpft.

Kapitel 7

Praxisbeispiele: Einführung von SAP S/4HANA

Welchen Aufwand ein Einführungsprojekt tatsächlich verursacht, können Sie aus unseren bisherigen theoretischen Erläuterungen nur schwer ableiten. Wir zeigen Ihnen deshalb in diesem Kapitel verschiedene Praxisbeispiele, die Ihnen die Einschätzung erleichtern.

Mit diesem Buch wollen wir Sie dabei unterstützen, die SAP-S/4HANA-Einführung in Ihrem Unternehmen vorzubereiten. Hierzu haben wir in den vorangegangenen Kapiteln dargelegt, welche Änderungen Sie in SAP S/4HANA im Vergleich zur SAP Business Suite erwarten und wie eine Installation bzw. Migration erfolgen kann. Die Erläuterungen haben wir teilweise anhand von Beispielen verdeutlicht.

In diesem Kapitel möchten wir Ihnen nun Praxisbeispiele für unterschiedliche Arten von SAP-S/4HANA-Implementierungsprojekten vorstellen. Den Anfang bildet mit Abschnitt 7.1 ein Implementierungsprojekt, bei dem eine zuvor genutzte Eigenentwicklung durch SAP S/4HANA abgelöst wurde. Eine besondere Anforderung dieses Projekts war, dass die Lösung in Stufen implementiert werden sollte und dass sich im Projektverlauf wesentliche organisatorische Änderungen ergeben haben, die zu berücksichtigen waren.

Abschnitt 7.2 beschreibt ein besonders herausforderndes Vorhaben, das eine Systemkonsolidierung zweier SAP-ERP-Systeme und den Wechsel auf die neue SAP-S/4HANA-Lösung zum Inhalt hatte. Im Kontext dieses Projekts spielen insbesondere Fragestellungen der Migration von Systemeinstellungen einerseits und der relevanten Daten andererseits eine wesentliche Rolle.

Anschließend stellen wir Ihnen in Abschnitt 7.3 ein Konvertierungsprojekt vor und erläutern dabei die relevanten Schritte, die im Rahmen des Wechsels von SAP ERP auf SAP S/4HANA umgesetzt werden mussten. Wir führen Sie so praxisnah durch den gesamten Migrationsprozess, damit Sie Besonderheiten, Werkzeuge und Problempotenziale einer SAP-S/4HANA-Konvertierung im Detail kennenlernen können.

Den Abschluss des Kapitels bildet Abschnitt 7.4 mit Projektbeispielen, in denen die Digitalisierungspotenziale von SAP S/4HANA in Verbindung mit SAP Fiori eindrucksvoll genutzt werden. Wir stellen Ihnen eine Barkassen-App vor, die auf der SAP-Barkasse basiert und die Verarbeitung für Gelegenheitsanwender elementar vereinfacht. Das zweite Beispiel ist die bpc Workflow-App, mit der Antragsverfahren sehr einfach digitalisiert und die Anwendungsoberflächen automatisch generiert werden können. Wir zeigen Ihnen hierfür mögliche Einsatzszenarien.

7.1 SAP-S/4HANA-Neuimplementierung mit einer Masterlösung am Beispiel des Bistums Limburg

Für Kunden, die bisher noch keine SAP-ERP-Lösung im Einsatz hatten, stellte sich bereits zum Zeitpunkt der Auslieferung der ersten Version von SAP S/4HANA die Frage nicht mehr, ob anstelle der neuen Lösung noch SAP ERP eingeführt werden sollte.

Anwenderfreundlichkeit

Die Anwenderfreundlichkeit von SAP S/4HANA, insbesondere für Gelegenheitsnutzer, ist ein wesentliches Argument dafür, eine SAP-Lösung einzuführen und nicht auf alternative Produkte zurückzugreifen. Die Eigenschaften der neuen Lösungsplattform motivieren zusammen mit den klassischen Argumenten für die Nutzung von SAP-Software nun auch Kunden, über den Einsatz dieser Lösung nachzudenken, für die SAP bisher noch nicht infrage kam. Diese klassischen Argumente sind beispielsweise Investitionssicherheit, Unabhängigkeit vom Hersteller bei der Implementierung und ein großer externer Expertenmarkt.

Die bekannte Stärke der SAP-ERP-Lösung, eigentlich alle denkbaren Anforderungen an ein ERP-System mit Standardfunktionen oder mit Werkzeugen für die Herstellung kundenindividueller Lösungsszenarien abbilden zu können, bleibt mit SAP S/4HANA erhalten. Der Weg, sich für das neue ERP-Produkt und für SAP-Software generell zu entscheiden, wird deshalb kürzer.

7.1.1 Besondere Anforderungen an die neue Lösung

Einstieg in die Digitalisierung

In unserem Referenzprojekt beim Bistum Limburg spielten alle genannten Aspekte eine wesentliche Rolle bei der Entscheidung dafür, eine selbst entwickelte ERP-Lösung bzw. eine heterogene Lösungslandschaft durch SAP S/4HANA abzulösen und die Grundlage für eine neue IT-Architektur zu schaffen. Auch für den Einstieg in die Digitalisierung wurde die Nutzung von SAP S/4HANA als elementarer Schritt gesehen. Vorangestellt wurde der

Implementierung ein Ausschreibungsverfahren, in dem die einzuführende Software und auch der Partner für die Implementierung ausgewählt wurden.

Die Entscheidung für SAP S/4HANA erfolgte schließlich auch aufgrund der Verfügbarkeit des *bpc-Kirchenmasters*, der projektbeschleunigend und qualitätssichernd eingesetzt wurde. Dieser beinhaltet unter anderem die folgenden Elemente:

- ein System-Template mit vorkonfigurierten Customizing-Einstellungen, entwickelten Apps, einem flexiblen Stammdatenkonzept, einem generischen Schnittstellentool und einem Werkzeug für den Haushaltsplandruck
- Dokumentationsmuster für Customizing und Entwicklungen
- Schulungsunterlagen und Vorlagen für Anwenderhandbücher
- Voreinstellungen für den SAP Solution Manager
- Vorlagen für fachliche und technische Konzeptionen

Von besonderer Bedeutung war die Anforderung des Bistums Limburg, im neuen ERP-System die folgenden Aspekte abbilden zu können:

- Für verschiedene organisatorische Elemente – Kirchengemeinden, Rentämter, Stiftungen, Friedhöfe, angeschlossene Körperschaften öffentlichen Rechts oder Kindergärten – muss die Erstellung von separaten Bilanzen und Gewinn- und Verlustrechnungen möglich sein.
- Es muss gleichzeitig möglich sein, jederzeit und einfach integrierte Berichte, Übersichten und Konsolidierungen über verschiedene Gruppen dieser organisatorischen Elemente aus dem System gewinnen zu können, um maximale Transparenz zu schaffen.
- Der Schritt in die Digitalisierung soll auf Basis der neuen Lösung gewährleistet werden. Dies bedeutet, dass einerseits Gelegenheitsanwender schnell und ohne besonderen Schulungsaufwand mit der Lösung arbeiten können müssen – und dass andererseits insbesondere Prozesse zur digitalen Rechnungseingangsverarbeitung mit der Lösung unterstützt werden sollen. Die Option zur elektronischen Aktenführung soll durch das Lösungsszenario ebenfalls bestehen.
- Das System sollte neben der digitalen Ausrichtung für die Haushaltsplanung als Produkthaushalt auch klassische analoge Funktionen bereitstellen – hierzu gehört insbesondere der Haushaltsplandruck als Auszug für einzelne organisatorische Einheiten wie auch als Gesamthaushaltsplan.

Referenzsysteme für schnelle und sichere Projekte

7.1.2 Lösungsmöglichkeiten mit SAP S/4HANA

In diesem Abschnitt möchten wir beschreiben, welche Lösungsmöglichkeiten SAP S/4HANA auf der Basis des bpc-Kirchenmasters bereitstellt, um die herausgestellten besonderen Anforderungen des Bistums Limburg zu erfüllen.

Differenzierte Organisationsstrukturen vs. Zentralisierung und Gesamttransparenz

Im Umfeld kirchlicher Kunden besteht generell die Notwendigkeit, dass scheinbar konkurrierende Anforderungen zeitgleich abbildbar sein müssen. Dies gilt auch für das Bistum Limburg:

- Zum Bistum gehören einerseits etwa 135 Gemeinden, die 11 Dekanaten zugeordnet sind. Die Gemeinden stellen genau wie der Pfarrfonds und der bischöfliche Stuhl Körperschaften dar, für die separate Abschlüsse zu erstellen sind.
- Teilweise den Gemeinden zugeordnet gibt es weiterhin nicht rechtsfähige, aber eigenständig bilanzierende Einheiten wie Kindergärten, Betriebe gewerblicher Art und Sondervermögen. Auch für sie muss eine Abbildung im System gefunden werden, die einen selbstständigen Abschluss erlaubt.
- Zentrale Funktionen für die dargestellten Einheiten im Kontext der Gemeinden sollen von zwei Rentämtern wahrgenommen werden. Zu den zentralen Funktionen gehören einerseits die Abwicklung des Mahn- und Zahlwesens und die Liquiditätssteuerung, aber auch die Pflege von Geschäftspartnerstammdaten und die Anlagenwirtschaft. Auch für diese zentralen Einheiten und die Abbildung von zentralen Funktionen war eine integrierte und wirtschaftliche Abbildungsform zu finden.
- Im Rahmen der Modernisierung implementiert das Bistum gegenwärtig Pfarreien neuen Typs, die als organisatorischer Zusammenschluss von Gemeinden zu betrachten sind. Die Organisation ist damit einer gewissen Dynamik unterworfen, die einfach im System abgebildet werden muss.
- Zentrale Verwaltungseinheit der Diözese Limburg ist das Bischöfliche Ordinariat – für dieses und die zugeordneten Einrichtungen und Stiftungen ist ebenfalls eine Abbildung im System zu finden, die eine Erstellung von Abschlüssen ermöglicht.

Abbildung 7.1 zeigt die relevanten organisatorischen Einheiten und welche betriebswirtschaftlichen Funktionen in den Einheiten benötigt werden.

Abbildung konkurrierender Anforderungen

organisatorische Elemente			Bilanz/GuV	Ergebnisrechnung	Anlagenpiegel	Liquiditätsbestand
Bistum	Kirchenstiftung	kirchl. Vereine/Verbände	●	●	●	●
Betriebe gewerblicher Art	Sondervermögen		●	●	●	●
Kindergärten	Tagungshäuser		○	●	○	●

Abbildung 7.1 Organisatorische Einheiten und benötigte Funktionen

Neben den dargestellten Anforderungen, Organisationseinheiten geeignet und separat abbilden zu können und zeitgleich betriebswirtschaftliche Prozesse zu zentralisieren, besteht weiterhin der Bedarf, bistumsweit Transparenz über kirchliche Leistungsbereiche zu schaffen oder Kollekten für bestimmte Zwecke über alle Organisationsbereiche hinweg auswerten, verwalten und abrechnen zu können.

Mit SAP S/4HANA ist die Abbildung dieser vielfältigen Anforderungen im Rahmen eines klaren Organisationsmodells möglich, ohne die Anwendung selbst komplex zu gestalten. Grundsätzlich könnte eine Umsetzung erfolgen, indem für jede Einheit, die einen separaten Abschluss benötigt, ein Buchungskreis angelegt wird – zentrale Auswertungen wären dann über Gruppierungen von Buchungskreisen bzw. über einen gemeinsamen Kostenrechnungskreis möglich. Nimmt man jedoch die Aspekte einer Zentralisierung und Bündelung von Funktionen und einer Dynamik in der Organisationsstruktur in die Überlegungen mit auf, stößt man bezüglich dieser Anforderungen an Grenzen.

Das mit SAP S/4HANA verbindliche neue Hauptbuch stellt dagegen einen Ansatz bereit, der es nicht erforderlich macht, bei der Zusammenführung von Gemeinden auch einen neuen Buchungskreis anzulegen, neue Sachkonten zu pflegen oder Daten zu migrieren: Bilanzen, Ergebnisrechnungen und Liquiditätsübersichten können mithilfe von Segmenten und Geschäftsbereichen dargestellt werden – die beliebige Gruppierbarkeit dieser Objekte ermöglicht zudem unterschiedliche Schnitte durch den gemeinsamen Datenwürfel bei der Erstellung von Berichten.

Komplexe Anforderungen abbilden

Abbildung 7.2 zeigt einen Lösungsansatz dafür, wie die organisatorischen Einheiten im SAP-S/4HANA-System abgebildet werden können.

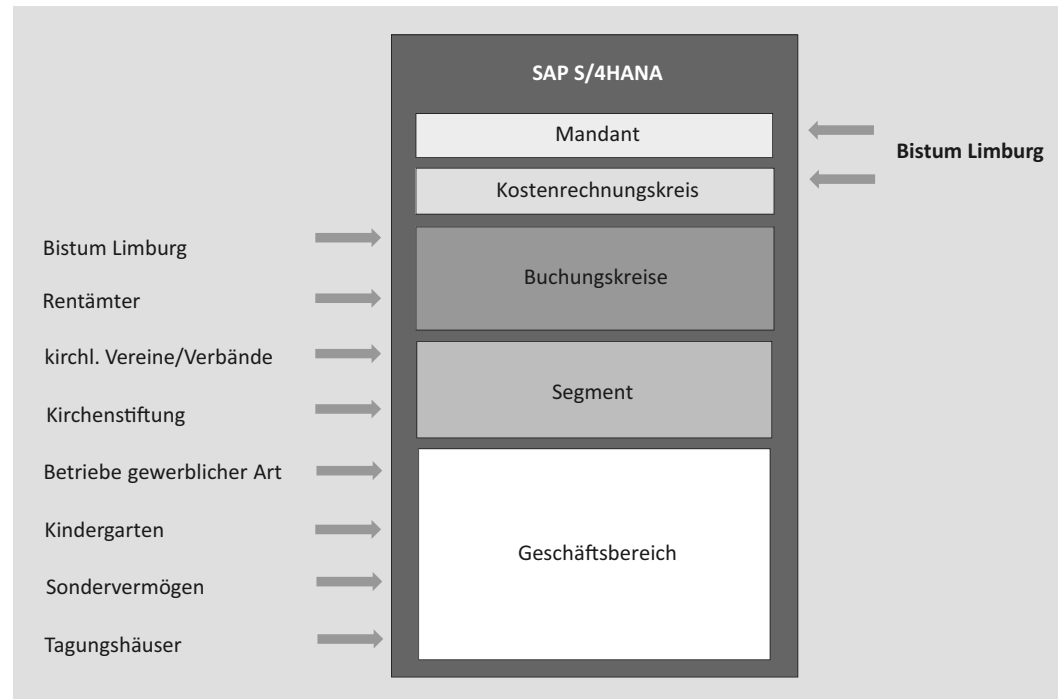


Abbildung 7.2 Abbildung der organisatorischen Einheiten in SAP S/4HANA

Zentrale Prozessunterstützung und dezentraler Einsatz

Der beschriebene Ansatz wird den Anforderungen insbesondere dann gerecht, wenn die folgenden Aspekte betrachtet werden:

- Geschäftspartner können kirchenweit gepflegt und angelegt werden und stehen damit jeder Kirchengemeinde für die Buchung zur Verfügung.
- Die zentrale Abwicklung von Buchhaltungsfunktionen wie Mahnen und Zahlen und Liquiditätssteuerung wird besser unterstützt und erfolgt in einheitlicher Form durch die Rentämter.
- Für alle beteiligten organisatorischen Einheiten und beliebige Gruppen dieser Einheiten können dieselben Berichte in einfacher Form genutzt werden – es sind etwa Dekanatsbilanzen mit und ohne zugeordnete Gemeinden genauso möglich wie Aufstellungen über alle Kindergärten im Bistum Limburg.
- Eine Zusammenlegung von Gemeinden führt nicht dazu, dass umfangreiche Migrationsprojekte notwendig werden.

Digitalisierung und Rechnungswesen-Workflow

Die Anforderung, den Einstieg in die Digitalisierung vor allem in der Rechnungsbearbeitung im Zusammenspiel zwischen zentralen und dezentralen Einheiten zu ermöglichen, kann mit SAP S/4HANA abgebildet werden, indem man die Lösung *SAP InvoiceManagement by OpenText* als in die SAP-Lösung integrierte Anwendung implementiert.

Diese Lösung fokussiert sich auf den Rechnungs-Workflow und nutzt dabei die Dokumentenmanagementsystem-(DMS-)Lösung Context Suite von OpenText – diese stellt neben der Rechnungsverarbeitung aber auch Funktionen für den Aufbau von Personal- oder Bauakten bereit. Abbildung 7.3 zeigt einen möglichen Ablauf für die digitalisierte Rechnungsverarbeitung.

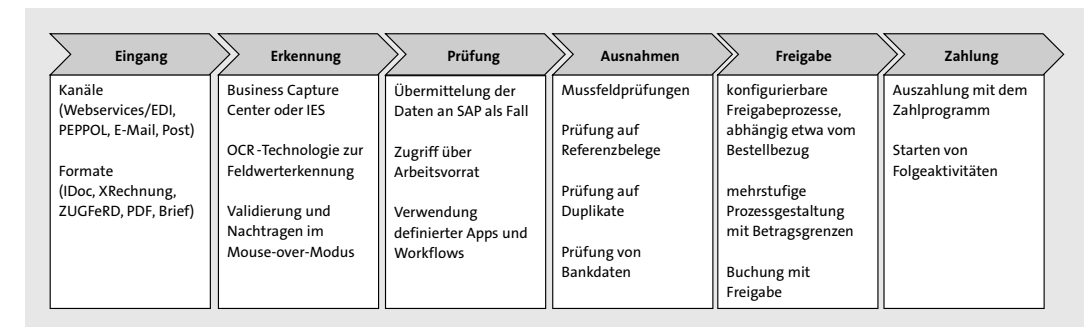


Abbildung 7.3 Ablauf der digitalisierten Rechnungsverarbeitung

Die Verarbeitung erfolgt in mehreren Schritten:

- Eingang der Rechnungen, die gescannt und im DMS abgelegt werden
- Transformation der Rechnung: Das Scanergebnis wird verarbeitet, und mithilfe einer OCR-Software werden Muster erkannt, validiert und Informationen zur Rechnung als Fall in der SAP-S/4HANA-Datenbank abgelegt.
- Abhängig von erkannten Werten und Mustern erfolgt eine weitere Prüfung, etwa durch das Anstoßen von Workflows. Im Rahmen dieser Workflows werden beispielsweise Kontierungsdaten per App ergänzt oder auch Stammdaten angelegt.
- Sofern definierte Ausnahmen aufgetreten sind – etwa durch Betragsüberschreitungen – können diese im Rahmen eines besonderen Monitorings bearbeitet und überwacht werden.
- Mit der Freigabe des Falls wird schließlich die Buchung erzeugt, sodass im letzten Schritt die Zahlung erfolgen kann.

Transparente
und digitalisierte
Rechnungs-
bearbeitung

Ein zentraler Aspekt des Projekts bestand darin, für die Phasen Prüfung und Komplettierung, Ausnahmereparatur und schließlich die Freigabe Werk-

zeuge zur Verfügung zu stellen, die vor allem den dezentralen Einsatz unterstützen.

Hierfür wurde eine Kontierungs-App entwickelt, die der Ergänzung von Kontierungsdaten dient und zur Bestätigung der sachlich-rechnerischen Richtigkeit verwendet werden kann. Abbildung 7.4 zeigt ein Beispiel für die Anwendung der Kontierungs-App.

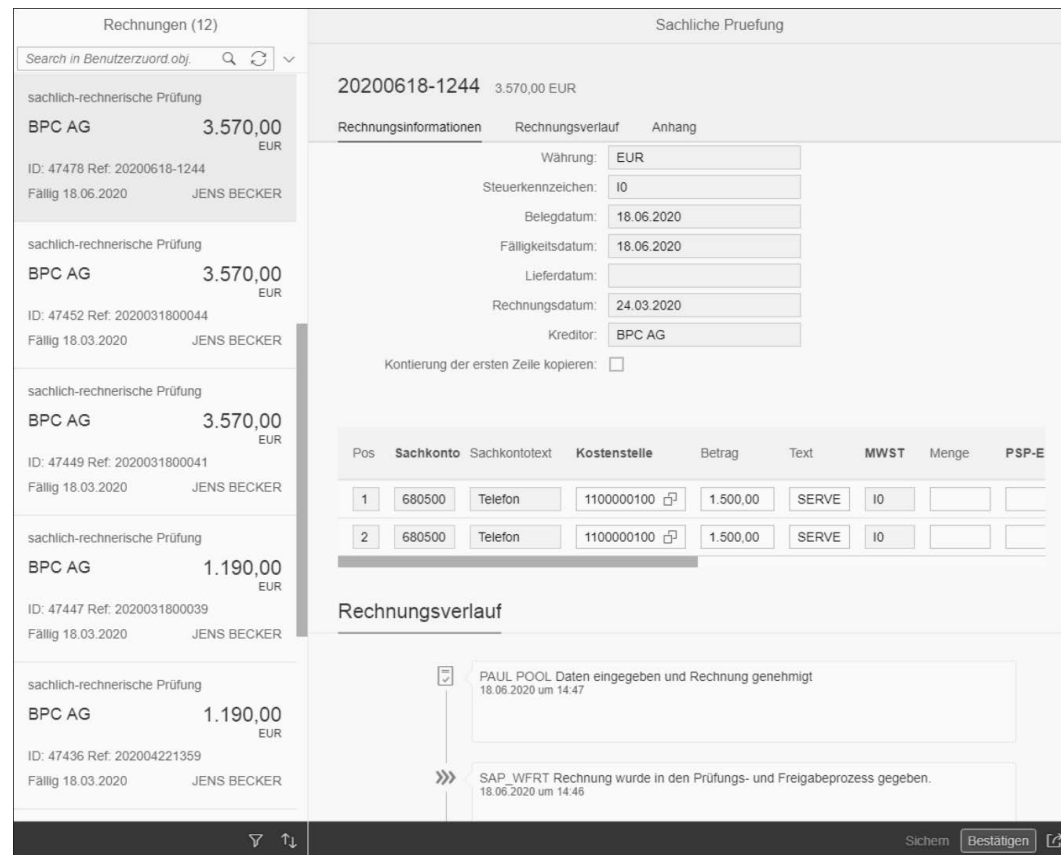


Abbildung 7.4 Beispiel für die Kontierungs-App

Apps für Kontierung und Freigaben

Durch die Nutzung der Freigabe-App werden insbesondere die Anwender unterstützt, die etwa im Rahmen des Freigabeprozesses nur ein Genehmigungskennzeichen setzen müssen. Wie Abbildung 7.5 zeigt, stehen dem Nutzer hierfür sowohl die im System erfassten Daten als auch die gescannte Rechnung in der Ansicht zur Verfügung.

Sofern eine Freigabe nicht erfolgen soll, wird der Fall durch den Rechnungswesen-Workflow an den entsprechenden Bearbeiter zur Korrektur von Daten übertragen.

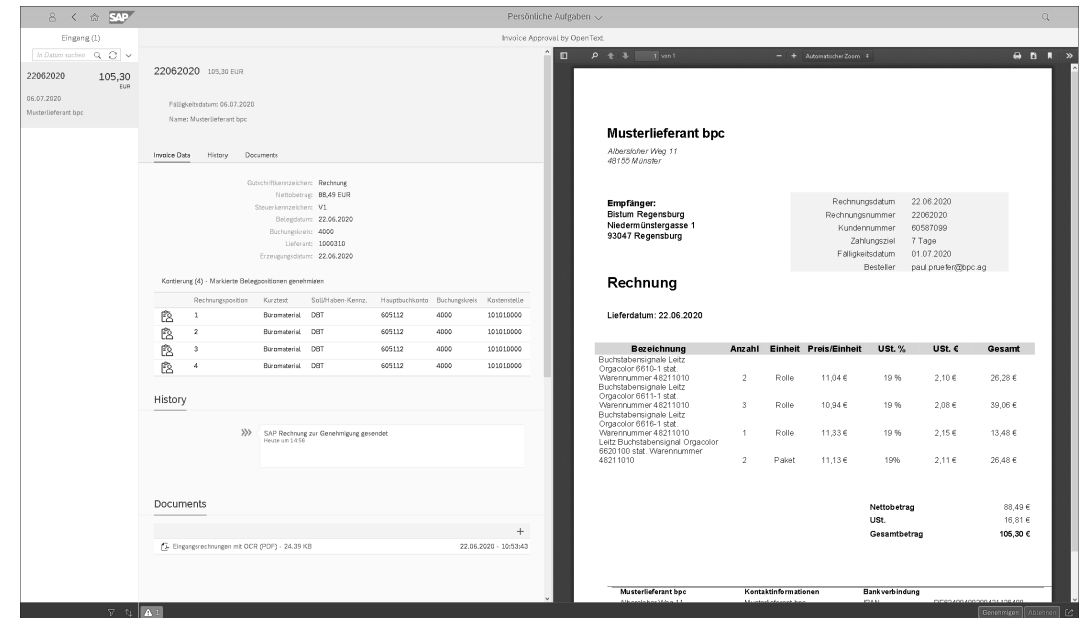


Abbildung 7.5 Beispiel für die dezentrale Freigabe per App

Als wesentliche Vorteile der neuen Lösung stellten sich für das Bistum Limburg folgende Aspekte heraus:

- einheitliche und transparente Gestaltung der Rechnungsverarbeitung
- Vermeidung von Skontoverlusten durch Monitoring-Möglichkeiten
- Vermeidung von doppelten Rechnungsbuchungen
- Möglichkeit zur Zentralisierung der Buchhaltungsaufgaben
- einfache Einbindung der beteiligten dezentralen Einheiten mithilfe der Kontierungs- und Freigabe-Apps

Haushaltsplanung und Haushaltsplandruck

Eine zentrale Anforderung des Bistums Limburg bestand in der Bereitstellung einer Lösung für die Haushaltsplanung, mit der einerseits ein mehrstufiger, integrierter Planungsprozess über verschiedene Organisationsstufen unterstützt werden kann und mit der andererseits der Druck eines Gesamt- wie auch beliebiger Teilpläne möglich ist.

Diese Lösung wurde mithilfe von *SAP Business Planning & Consolidation* (SAP BPC) für SAP S/4HANA mit der bpc-App für den Haushaltsplandruck umgesetzt.

Neue Planungsfunktionen mit SAP BPC

SAP BPC bietet eine konsistente Sicht auf den Planungsprozess und ermöglicht es, strukturierte Planungsprozesse über mehrere Stufen und über jeweils relevante Datenausschnitte zu definieren. So ist es beispielsweise möglich, zunächst zentral für alle Kindergärten des Bistums bestimmte Werte zu planen, um dann dezentral Feinplanungen vorzunehmen, diese automatisch zu konsolidieren und schließlich zentral zu finalisieren. Der Planungsprozess selbst unterliegt dabei einem Monitoring – es ist somit nachvollziehbar, an welcher Stelle in der Organisation gegenwärtig die Planung durchgeführt wird (siehe Abbildung 7.6).

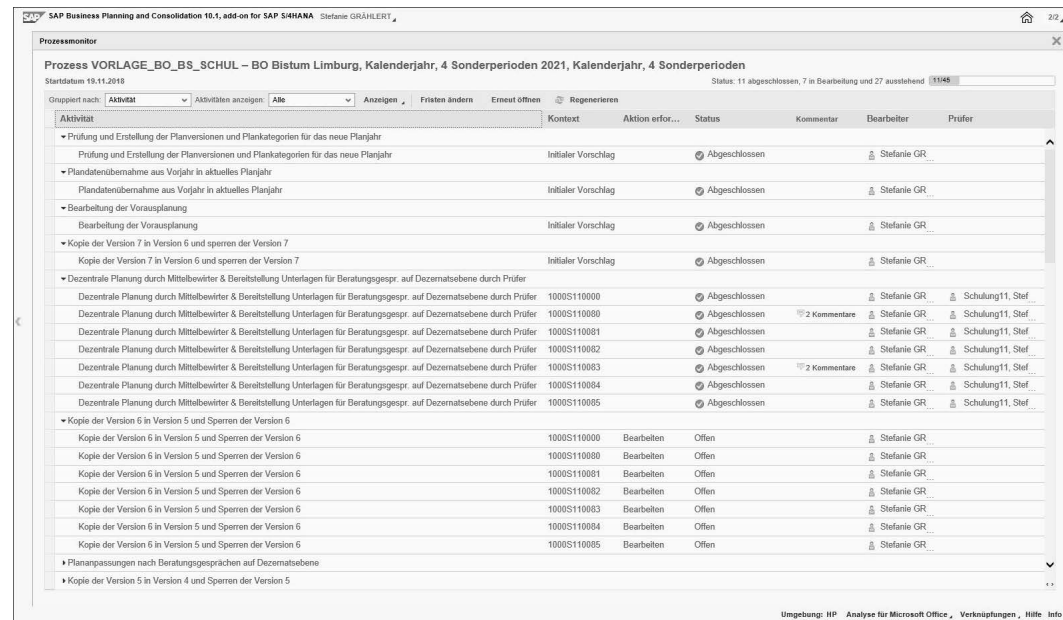


Abbildung 7.6 Monitoring des Planungsprozesses

Die Planung selbst erfolgt beispielsweise mit SAP Analysis for Microsoft Office für die zentralen Anwender und SAP Lumira für die dezentralen Anwender und kann in für den Anwender gewohnten Excel-Masken bzw. Weboberflächen erfolgen. Funktionen wie automatische Verteilungen, das Kopieren von Daten oder die automatische Aggregation von Daten sind für den Anwender ebenfalls verfügbar. Abbildung 7.7 zeigt ein Beispiel für eine Planungsmaske, die dem Benutzer zur Verfügung gestellt wird.

Die Planungsanwendungen des Bistums basieren auf einem vollständig in SAP S/4HANA integrierten Planungsmodell mit einer eigenen Plandatenpersistenz. Stamm- und Bewegungsdaten aus dem Finanzwesen und Controlling können direkt innerhalb der Planungsprozesse genutzt werden.

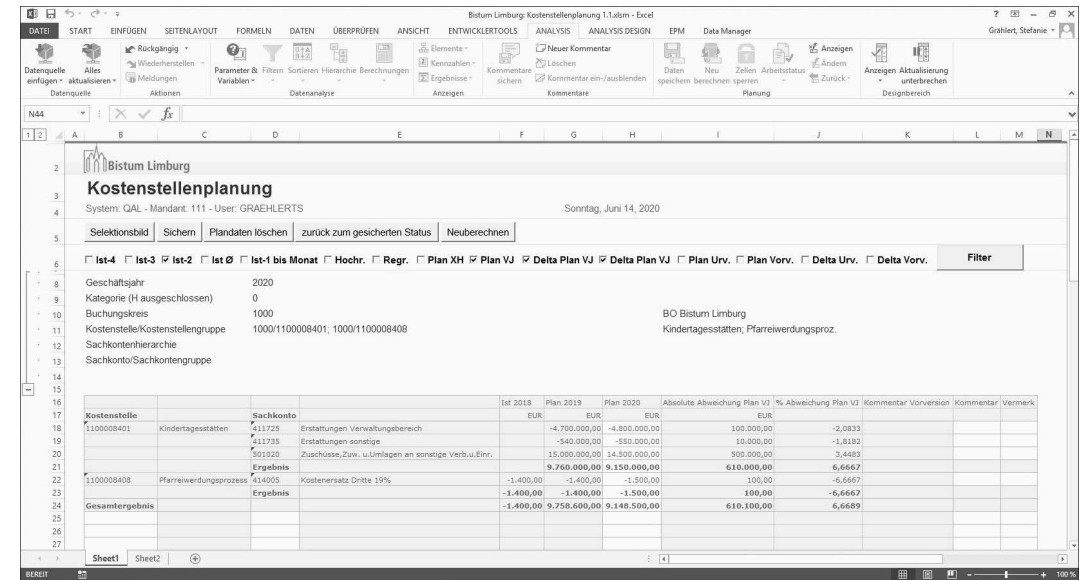


Abbildung 7.7 Planungsmaske mit SAP Analysis

Dem Anwender stehen damit zur Unterstützung des Planungsprozesses Daten in Echtzeit zur Verfügung – sie werden im SAP Embedded BW vorgehalten.

Für den Druck des Haushaltsplans wurde ein von bpc entwickeltes Werkzeug zum Generieren von Haushaltsplänen verwendet. Es ermöglicht unter anderem die Abbildung der folgenden Anforderungen:

- Erzeugung eines Gesamtergebnisplans
- Erzeugung von Teilergebnisplänen je Element bezüglich unterschiedlicher Hierarchieebenen (etwa Produktbereichen, -gruppen oder Produkten)
- Erstellung von Investitionsmaßnahmenplänen (gesamt und je Element unterschiedlicher Hierarchieebenen)
- Einbinden von Texten und Grafiken in die jeweiligen Pläne

Die Erzeugung der Ausdrücke erfolgt mithilfe von Microsoft Word – während die Lösung selbst in ABAP/OO realisiert wurde. Abbildung 7.8 zeigt einen Auszug aus dem gedruckten Haushaltsplan des Bistums Limburg mit Beispieldaten, Abbildung 7.9 zeigt die Fiori-App für den Aufruf.

Haushaltsplan-
druck für beliebige
Hierarchieebenen

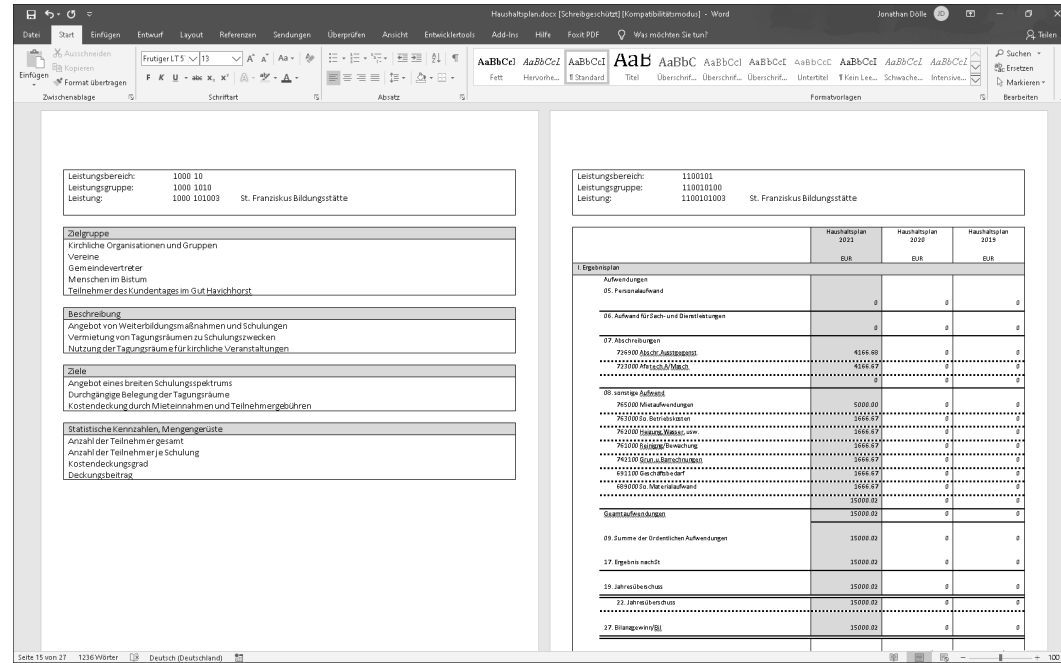


Abbildung 7.8 Ein Auszug aus dem Haushaltsplan in Word

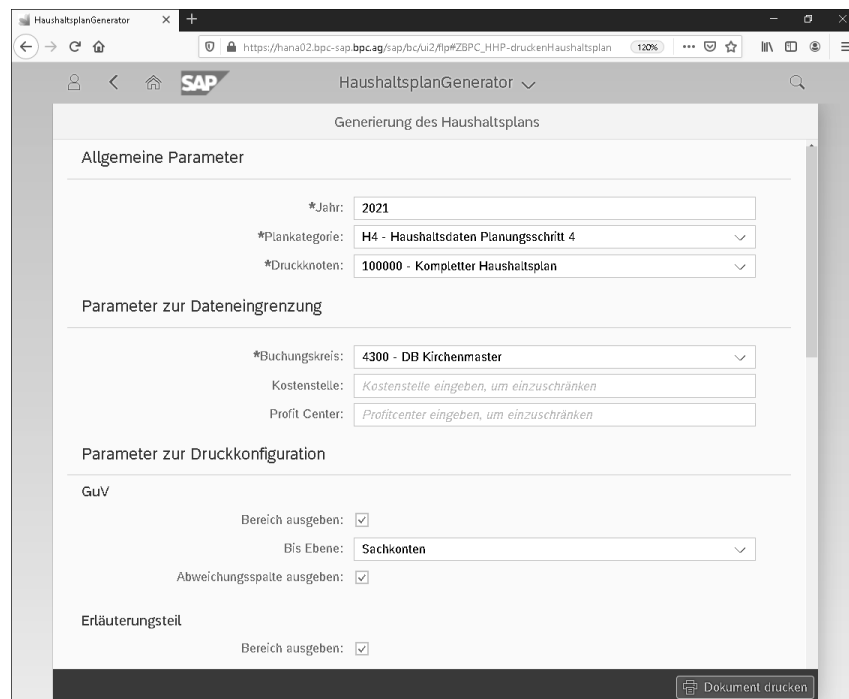


Abbildung 7.9 Die Fiori-App für die Erstellung des Haushaltsplans

Die Lösung selbst kann mithilfe von Konfigurationstabellen und von Apps zur Konfiguration parametrisiert werden. Der Ausdruck kann ebenfalls über eine App initiiert werden.

Die SAP-Einführung beim Bistum Limburg war mehr als ein rein technischer Umstellungsprozess in der Buchhaltung. Es beinhaltete vielmehr eine Neuausrichtung des Planungswesens auf inhaltliche Schwerpunkte und den Einstieg in die Digitalisierung von Arbeitsprozessen, unter anderem durch die künftig elektronische Eingangsrechnungsbearbeitung.

7.2 Systemkonsolidierung am Beispiel der ELKB

Als Beispiel für eine Systemkonsolidierung verbunden mit einem hybriden Einführungskonzept möchten wir Ihnen das SAP-S/4HANA-Implementierungsprojekt bei der Evangelisch-Lutherischen Kirche in Bayern (ELKB) vorstellen.

Die ELKB setzte zu Projektstart zwei SAP-ERP-Systeme produktiv ein. Für die Landeskirche wurde seit vielen Jahren ein SAP-ERP-System genutzt, in dem viele kundenindividuelle Anpassungen umgesetzt und auch zahlreiche Modifikationen vorgenommen wurden. Die Wartung dieses Systems war daher sehr aufwendig. Für die kirchlichen Gemeinden und dazugehörigen Körperschaften wurde ein SAP-ERP-System (VfKG-System: Verfahren für die Kirchlichen Gemeinden) verwendet. Es basierte auf dem bpc-Kirchenmaster, verwendete für die Planungsfunktionen, Berichte und Barkassenprozesse im dezentralen Bereich aber eine Portalapplikation, die mit dem ERP-System gekoppelt war. Dieses System wurde seit 2016 eingesetzt, sodass der Datenbestand noch relativ gering war. Gleichzeitig befand sich die SAP-Lösung für die Gemeinden inmitten des Roll-outs auf weitere Dekanatsbezirke.

Aufgabe des Projekts war die Zusammenführung beider Systeme zu einem gemeinsamen SAP-S/4HANA-System einschließlich einer geeigneten Datenmigration. Hierfür musste ein gemeinsamer Migrationspfad entwickelt, abgestimmt und umgesetzt werden. Zusätzlich sollte das Projekt zur Umstellung auf SAP S/4HANA in den laufenden Roll-out der VfKG-Lösung integriert werden. In den folgenden Abschnitten möchten wir dies genauer darstellen.

7.2.1 Besondere Anforderungen im Projekt

Für den Aufbau des neuen Systems waren verschiedene Rahmenbedingungen zu berücksichtigen, die die Beantwortung einer der Hauptfragestellungen

Zwei SAP-ERP-Systeme müssen zusammengeführt werden

gen des Projekts maßgeblich beeinflusst haben: Soll eines der Systeme konvertiert und anschließend die Daten des anderen Systems übernommen werden, oder soll ein gemeinsames System neu aufgebaut werden?

Prüfungsanforderungen

Der Datenbestand auf dem VfKG-System war zwar relativ jung, doch befanden sich auf dem System noch offene Geschäftsjahre, für die ein Abschluss, der insbesondere für die Anlagenwirtschaft erforderlich ist, zum Projektstart nicht möglich war. Für diese Geschäftsjahre musste jedoch gewährleistet werden, dass verschiedene Prüfungsanforderungen (RPA, Wirtschaftsprüfer, Steuerprüfung) bei der Umstellung befriedigt werden konnten und hierfür unterschiedliche Anforderungen bezüglich des Datenzugriffs abzubilden waren. Zudem waren die Prüfungen teilweise nicht zeitlich beeinflussbar. Diese Anforderungen erschwerten die Umsetzung des scheinbar offensichtlichen Ansatzes, das jüngere System zu konvertieren und zur Basis der konsolidierten Lösung zu machen.

Erhalt des Datenbestandes

Für den Fall, dass eine Konvertierung des jüngeren VfKG-Systems nicht als Lösung in Betracht kam, entstand eine neue Anforderung: Bei der Umstellung sollten die Voraussetzungen dafür geschaffen werden, dass einerseits keine Doppelerfassung von Vorgängen in zwei Systemen notwendig werden und andererseits vollständige historische Werte (etwa Vorjahreswerte im Bereich Controlling) im neuen System verfügbar waren. Auch das Nachholen von Abschlüssen sollte im neuen System möglich sein. Die Verfügbarkeit des vollständigen Datenbestandes im neuen SAP-S/4HANA-System war auch vor dem Hintergrund der Akzeptanz bei den Anwendern, die erst vor Kurzem die Einführung der Lösung SAP ERP Central Component (SAP ECC) vollzogen hatten, ein wesentlicher Aspekt.

Prüfung der Erweiterungen

Eine weitere Anforderung bestand in den unterschiedlichen Ständen der Systeme, insbesondere hinsichtlich vorgenommener Erweiterungen und Modifikationen des landeskirchlichen Systems. Die Prüfung, ob die Erweiterungen auch weiterhin notwendig sind bzw. wie eine Ablösung dieser Funktionen erfolgen kann, stellte einen wesentlichen Projektaufwand dar, der auch die eigene Organisation der ELKB intensiv belastet hat. Es galt deshalb, einen Pfad für die Umstellung zu gestalten, der alle genannten Aspekte berücksichtigt.

Ein einheitliches System

Eine einheitliche Ausgestaltung der Funktionen und Prozesse zwischen der Landeskirche und der gemeindlichen Ebene als Basis eines einheitlichen, möglichst effizient und effektiv ausgestalteten und betreibbaren Systems war eine weitere zentrale Herausforderung im Projekt.

Auf der einen Seite ist eine derartige Konsolidierung die Basis für eine gemeinsame Systemnutzung, auf der anderen Seite bedeutete sie, gerade erst

eingeführte (oder auch seit Langem etablierte) Prozesse überdenken und unter Umständen anpassen zu müssen. Mit Blick auf externe Einflussfaktoren – etwa die Digitalisierung und Modernisierung der kirchlichen Verwaltung, den demografischen Wandel und den damit verbundenen Arbeitskräftemangel – galt es, bei der Systemumstellung und -konsolidierung die Organisation und ihre Mitarbeiter geeignet mitzunehmen und zu begleiten.

7.2.2 Lösungsansätze

Die ELKB entschied sich aufgrund der dargestellten Restriktionen für einen Projektansatz, der den Aufbau des neuen, gemeinsamen Systems in mehreren Umstellungsstufen realisiert. In einer ersten Projektphase wurde auf Basis des bpc-Kirchenmasters und der gemeinsamen Konzeption von VfKG und Landeskirche das gemeinschaftliche Grundsystem aufgebaut. In einer zweiten Phase (Release 1) wurde es um die spezifischen Prozesse, Funktionen und Daten von VfKG erweitert, bevor in der dritten Phase (Release 2) die spezifischen Funktionen, Prozesse und Daten der Landeskirche berücksichtigt wurden.

Im Rahmen des Release 1 wurde ein zweistufiger Go-live mit Pilotierung und Roll-out vorgenommen: Die Pilotierung umfasste die Buchungskreise des Dekanatsbezirks Augsburg, der Roll-out dann die Umstellung von SAP ECC auf SAP S/4HANA für die übrigen Dekanatsbezirke. Abbildung 7.10 zeigt dies in einer Übersicht.

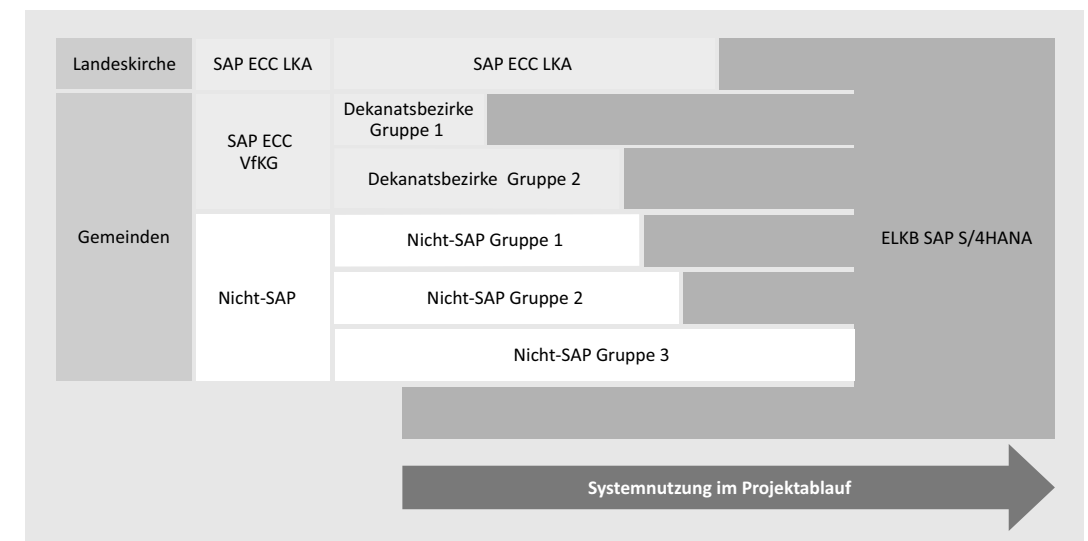


Abbildung 7.10 Projektplan ELKB

Umstellung in mehreren Stufen

Hinsichtlich der Fragestellung Konvertierung oder Neuaufbau entschied man sich dementsprechend grundsätzlich für das Konzept, ein neues System zu implementieren.

Neuaufbau im hybriden Ansatz

Der Neuaufbau des Systems erfolgte jedoch mit einem hybriden Ansatz: Es wurden ein neues SAP-S/4HANA-System installiert, Entwicklungen und Customizing-Einstellungen aus dem bestehenden VfKG-System selektiv mithilfe von Transportaufträgen übernommen und um zusätzliche Einstellungen ergänzt.

Durch die Verwendung dieses Ansatzes ergaben sich besondere Herausforderungen im Zuge der Datenmigration: Die Anwender hätten es nicht akzeptiert, wenn ihnen zusätzlicher Aufwand durch die Migration entstanden wäre. Um eine doppelte Datenhaltung und doppelte Erfassungsaufwände zu verhindern, wurde beschlossen, die Daten des relativ jungen VfKG-Systems mittels einer Komplettmigration in das neue System zu transferieren. Dies kam dem Ergebnis einer Konvertierung nahe. Dieses Vorgehen implizierte, dass alle Belege, die bis zum jeweiligen Stichtag vor der Übernahme im alten System gebucht wurden, beim Go-live zu den entsprechenden Stichtagen übernommen werden mussten und dabei auch Historien – wie etwa zur Abrechnung von Innenaufträgen oder zur Verarbeitung elektronischer Kontoauszüge, zu berücksichtigten waren.

Komplexe Datenmigration

Der Prozess der Datenmigration war damit aufgrund der Zweiteilung, der Unterjährigkeit und des dualen Betriebs des neuen und des alten Systems detailliert zu konzeptionieren und nicht an allen Stellen mit den SAP-Standardwerkzeugen zu bewerkstelligen. Insbesondere war zu vermeiden, dass Benutzer parallel in zwei Systemen arbeiten sollten.

Festlegung des Migrationsvorgangs

Eine große Herausforderung der Vollmigration stellte die technische Umsetzung dar, weil diverse Abhängigkeiten zwischen den zu migrierenden Objekten bestehen und die Übernahme von kompletten Geschäftsjahren an vielen Stellen nicht mit Standardwerkzeugen durchzuführen ist. Zu klären waren damit die Fragestellungen:

- Welche Daten sollen übernommen werden?
- Welches Werkzeug ist für die Übernahme der jeweiligen Daten geeignet?

Abbildung 7.11 zeigt hierzu wesentliche Fragestellungen in der Form eines morphologischen Kastens.

Übernahme historischer Daten	alle historischen Daten (Einzelposten)	aggregierte historische Daten	keine historischen Daten
Datenumfang	alle Einzelposten		keine stornierten Belege
Umgang mit Ausgleichen	Ausgleiche 1:1 nachbuchen		Ausgleiche nachgelagert als separate Ausgleichsbelege buchen
Technik	DB-Insert		Nachbuchen von Belegen
Werkzeug	SLT	SAP Data Services	SAP-integriert (Migrationscockpit, BAPI...)

Abbildung 7.11 Morphologischer Kasten für die Migrationsstrategie

Hinsichtlich der historischen Daten war zu klären, ob eine Übernahme aller historischen Daten als Einzelposten erforderlich ist, ob eine aggregierte Übernahme erfolgen soll oder ob auf historische Daten verzichtet werden kann.

Übernahme historischer Daten

Hinsichtlich der Werkzeugauswahl und der konkreten Übernahmetechnik spielten auch die Gewährleistungsverpflichtungen von SAP eine entscheidende Rolle: Eine Migration von Belegen mittels Datenbank-Insert lässt die Gewährleistung erlöschen, sodass man sich dafür entschied, insbesondere für die Belegübernahme bekannte Werkzeuge wie IDocs und BAPIs zu verwenden. Kundeneigene Programme mit BAPIs spielten insbesondere beim nachträglichen Ausgleichen von Belegen eine besondere Rolle: Die Standardwerkzeuge von SAP für die Datenmigration einschließlich des Migrationscockpits bieten keine Möglichkeiten zum Belegausgleich bei der Datenmigration. Im Projekt wurde schließlich die Entscheidung getroffen, die Migration entlang der Variante »Nachbuchen mit eingeschränkter Vollmigration« als Lösung zu verfolgen (siehe Abbildung 7.12).

Übernahme historischer Daten	alle historischen Daten (Einzelposten)	aggregierte historische Daten	keine historischen Daten
Datenumfang	alle Einzelposten		keine stornierten Belege
Umgang mit Ausgleichen	Ausgleiche 1:1 nachbuchen		Ausgleiche nachgelagert als separate Ausgleichsbelege buchen
Technik	DB-Insert		Nachbuchen von Belegen
Werkzeug	SLT	SAP Data Services	SAP-integriert (Migrationscockpit, BAPI...)

Abbildung 7.12 Umsetzungsvariante für Nachbuchungen

Nachbuchen von Belegen Bei diesem Ansatz werden alle nicht stornierten Belege ins neue System übernommen und die Ausgleiche nicht 1:1 nachvollzogen, sondern als separate Ausgleichsbelege im neuen System nachgebucht. Durch dieses Nachbuchen der Ausgleiche in Form separater Ausgleichsbelege wurde die Umsetzung der Ausgleichsbuchungen vereinfacht.

Zu Bedenken waren aber weiterhin die folgenden Aspekte:

- Die Systemeinstellungen mussten so getroffen werden, dass ein Nachbuchen der alten Belege auch im neuen System möglich war.
- Buchungen aus Nebenbüchern (FI-AA, TR) mussten dort nachgebucht werden, um die Konsistenz sicherzustellen.
- Der Buchungsstoff musste umfangreich angereichert werden, weil infolge geänderter Belegaufteilungsregeln inhaltliche Änderungen entstanden sind.

Bei der Auswahl des Szenarios wurden neben den Fachbereichen insbesondere auch rechnungsprüfende Bereiche und Revisoren mit einbezogen. Begleitend während der gesamten Konzept- und Umsetzungsphase der Migration galt es, gerade diesen Einbezug intensiv auszugestalten und überall dort in einen detaillierten Dialog einzusteigen, wo aufgrund technischer und inhaltlicher Restriktionen Abweichungen zwischen altem und neuem System unvermeidbar waren.

Auswahl der Werkzeuge

Kriterien zur Werkzeugauswahl Grundsätzlich wichtig für die Auswahl geeigneter Werkzeuge für die Datenübernahme waren die Eigenschaften bezüglich der folgenden Kriterien:

- Vollständigkeit der Daten (K.-o.-Kriterium)
- Aufwand für die Einrichtung und Bedienung des Werkzeugs (einschließlich der Datenaufbereitung)
- Nachvollziehbarkeit und Protokollierung der Übernahme
- Verfügbarkeit einer Testverarbeitung
- Geschwindigkeit bei der Verarbeitung
- Nähe zum SAP-Standard

Tabelle 7.1 zeigt ausgewählte Werkzeuge für die Migration, deren Einsatz im Projekt geprüft wurde.

Werkzeug	Wesentliche Eigenschaften/Aspekte
SAP Kopierwerkzeuge	Für einige Stammdatenobjekte, etwa Sachkonten, bieten SAP ERP und SAP S/4HANA Standardwerkzeuge an, mit deren Hilfe die Daten sehr einfach in Form von Textdateien heruntergeladen und in anderen Systemen wieder hochgeladen werden können.
SAP S/4HANA Migration Cockpit	Zur Migration von Stamm- und einigen Bewegungsdaten, aber auch für eigene Objekte kann das SAP S/4HANA Migration Cockpit genutzt werden. Wir beschreiben dieses ausführlich in Abschnitt 8.4.3, »Neue Werkzeuge zur Datenmigration«.
Legacy System Migration Workbench (LSMW)	Obwohl die LSMW für die Migration nicht mehr von SAP empfohlen wird, kann diese Option weiterhin verwendet werden.
ALE (Application Link Enabling)/IDoc	IDocs können im Rahmen der Migration genutzt werden, um Daten aus dem SAP ECC ins neue SAP S/4HANA-System zu verteilen. Die Verteilung der Daten erfolgt dabei durch eine direkte Kommunikation auf Serverebene und ist nicht auf Datei- oder sonstige Exporte angewiesen. Ein Mapping ist bei der Verarbeitung der IDocs im Zielsystem ebenfalls möglich.
Programmierung mit Aufruf von SAP-Standard-BAPIs	Die Verwendung von BAPIs im Rahmen von Eigenentwicklungen kann insbesondere dann sinnvoll sein, wenn Daten aus dem sendenden System gelesen und im empfangenden System direkt verbucht werden sollen.
Programmierung mit direktem Schreiben in SAP-Tabellen	Bei vergleichsweise »isolierten« Daten (keine Verteilung über zu viele SAP-Tabellen) kann das direkte Schreiben in die SAP-Tabellen des Zielsystems als Lösungsansatz zugrunde gelegt werden. Auch hier ist jeweils der Ansatz zu prüfen, die Daten im sendenden System zu lesen und unmittelbar im empfangenden System einzubuchen.
Transportaufträge	Es besteht die Möglichkeit, vollständige Tabelleninhalte über Transportaufträge zu kopieren.

Tabelle 7.1 Werkzeuge für die Migration

**Direkte
Übertragung von
Daten mit ALE**

Die Form der Datenextraktion und der Umsetzung in ein importfähiges Format des jeweiligen Werkzeugs spielte eine wesentliche Rolle im Entscheidungsprozess. Insbesondere um ungewollte Veränderungen an den zu migrierenden Daten zu verhindern, wurde überall die Möglichkeit einer Onlineübertragung der Migrationsdaten geprüft. Dies war zum Projektzeitpunkt insbesondere für die Szenarien »Nutzung von IDocs« und den individuellen Programmierungen möglich – inzwischen bietet auch das SAP S/4HANA Migration Cockpit entsprechende Möglichkeiten, Daten beim Sender zu lesen und beim Empfänger zu verbuchen. Aufgrund der einfachen Einsatzmöglichkeit der IDoc-Nutzung verbunden mit der Leistungsstärke der Lösung hinsichtlich Datensammlung und Geschwindigkeit spielte die Nutzung dieser Technologie eine wesentliche Rolle. Tabelle 7.2 enthält einen Ausschnitt der migrierten Objekte und der jeweils genutzten Technik.

Objekt	Technik
CO-Stammdaten	IDoc
Set-Hierarchien	Kopieren mit GR57/GR58
Projekte/PSP-Elemente	IDoc
Anlagen im Bau zu Investitionsaufträgen	LSMW
Abrechnungsvorschriften	Tabellenkopien
Sachkonten	FS15/FS16
Geschäftspartner	Programmierung mit Aufruf von SAP-Standard-BAPIs
SEPA-Mandate	Programmierung mit Aufruf von SAP-Standard-BAPIs
Anlagen (nicht AiB)	LSMW
TRM-Stammdaten	Programmierung mit Aufruf von SAP-Standard-BAPIs
Geschäfte TRM	Programmierung mit Aufruf von SAP-Standard-BAPIs
Banken für ZV	Migrationscockpit
CO-Zyklen	Transportauftrag
Belege FI inkl. CO/EC	IDoc

Tabelle 7.2 Ausschnitt von Migrationstechniken

Objekt	Technik
Anfangsbestand AA	LSMW
Belege FI-AA integr.	Programmierung mit Aufruf von SAP-Standard-BAPIs
Abrechnung Investitionsaufträge und PSP-Elemente	Transaktionen KO8G, CJ8G
Sonderposten Zu-/Abgänge, Umbuchung	Programmierung mit Aufruf von SAP-Standard-BAPIs
AfA, Sonderpostenauflösung	AfA-Läufe
TRM-Finanzströme	Programmierung mit direktem Schreiben in SAP-Tabellen
Belege TRM (Nebenbuch)	Programmierung mit direktem Schreiben in SAP-Tabellen
ELKOs/Belege FI-BL	DB-Insert
Ausgleiche	Programmierung mit Aufruf von SAP-Standard-BAPIs
Dauerbuchungsurbelege	Programmierung mit direktem Schreiben in SAP-Tabellen
CO-Plandaten	Programmierung mit Aufruf von SAP-Standard-BAPIs
Änderungsbelege	DB-Insert

Tabelle 7.2 Ausschnitt von Migrationstechniken (Forts.)

Für die Abstimmung der Migrationsergebnisse wurden die Standardberichte zugrunde gelegt, die von SAP üblicherweise für Migrationsprojekte empfohlen werden:

- die Bilanz/GuV (Report RFBILAOO)
- der Summenbericht für Kostenstellen (Transaktion S_ALR_87013611)
- Auftrag: Ist/Plan/Abweichung (Transaktion S_ALR_87012993)
- die Saldenliste Sachkonten (Report RFSSLD00)
- die Einzelpostenliste Hauptbuch (Report RFSOPO00)
- das Belegkompaktjournal (Report RFBELJ00)
- das Anlagengitter (Report RAGITT_ALV01)
- den Abschreibungslauf für geplante Abschreibungen (RAHAFA_ALV01)

**Abstimmung
der Migrations-
ergebnisse**

- die Kreditoren-Umsätze (Report RFKUMLOO)
- die Kreditoren Offene Posten Liste (Report RFKEPLOO)
- die Debitoren-Umsätze (Report RFDUMLOO)
- die Debitoren Offene Posten Liste (Report RFDEPLOO)
- die Debitoren Dauerbuchungs-Urbelege (Report RFDAUB00)

Der Vergleich der Berichte für die migrierten Buchungskreise im VfKG und SAP-S/4HANA-System war ein wichtiger Indikator für eine erfolgreiche Datenübernahme. Die Dokumentation des Vorgehens und die Ergebnisse der Prüfung wurden zum Nachweis für Revisoren und Prüfer detailliert abgestimmt, um auch in diesem Kontext alle Anforderungen und eine gesetzeskonforme Buchführung sicherstellen zu können.

Digitalisierung als Herausforderung

Zur Beschreibung der Herausforderung, die Systemumstellung und -konsolidierung mit der Digitalisierung und Modernisierung zu verknüpfen und dabei die Organisation und ihre Mitarbeiter geeignet mitzunehmen und zu begleiten, möchten wir noch zwei Aspekte hervorheben:

Die Einführung einer vollständig digital abgebildeten elektronischen Rechnungseingangslösung mithilfe der SAP-Standard-Lösung *OpenText Vendor Invoice Management for SAP Solutions* (VIM), verknüpft mit einer weitreichenden Einführung von SAP Fiori und mit der Perspektive eines ganzheitlichen digitalen Dokumentenmanagements wurden als zentrale Bausteine in das Projekt integriert. Auf diese Weise wurde ein schrittweiser, für die Benutzer und die Organisation als Ganzes zu verkraftender Einstieg in die Digitalisierung und konkret die digitale Vorgangsbearbeitung umgesetzt.

Das gesamte Projektvorhaben wurde von einem intensiven Changemanagement begleitet. Hierfür wurde zu Beginn des Vorhabens eine detaillierte Analyse aller Prozesse, Funktionen, Organisationseinheiten und Personen vorgenommen, die von Änderungen durch das Projekte betroffen waren. Darauf basierend wurden pro Auswirkung und betroffener Einheit entsprechende Maßnahmen ausgearbeitet und umgesetzt – dazu gehörten unter anderem Informationsmaßnahmen, Schulungen und Organisationsanpassungen.