



Was wir vorhersagen, soll auch eintreffen!



## Software Entwicklung mit SAP HANA //

*Sven Broeske*

## Software Entwicklung mit SAP HANA

Im SAP HANA Kontext existiert eine Vielzahl an Werkzeugen und Skriptsprachen zur Realisierung anspruchsvoller Softwarelösungen. Damit ein Grundverständnis aufgebaut werden kann und eine effektive Nutzung der gebotenen Hilfsmittel möglich ist, wird in diesem Dokument auf einen Großteil der zur Verfügung stehenden Werkzeuge und Skriptsprachen eingegangen. Damit eine Einordnung der beleuchteten Themen ins komplexe SAP HANA Universum möglich ist, wird in Abb. 1 ein grundlegender Überblick über das SAP HANA Universums gegeben.

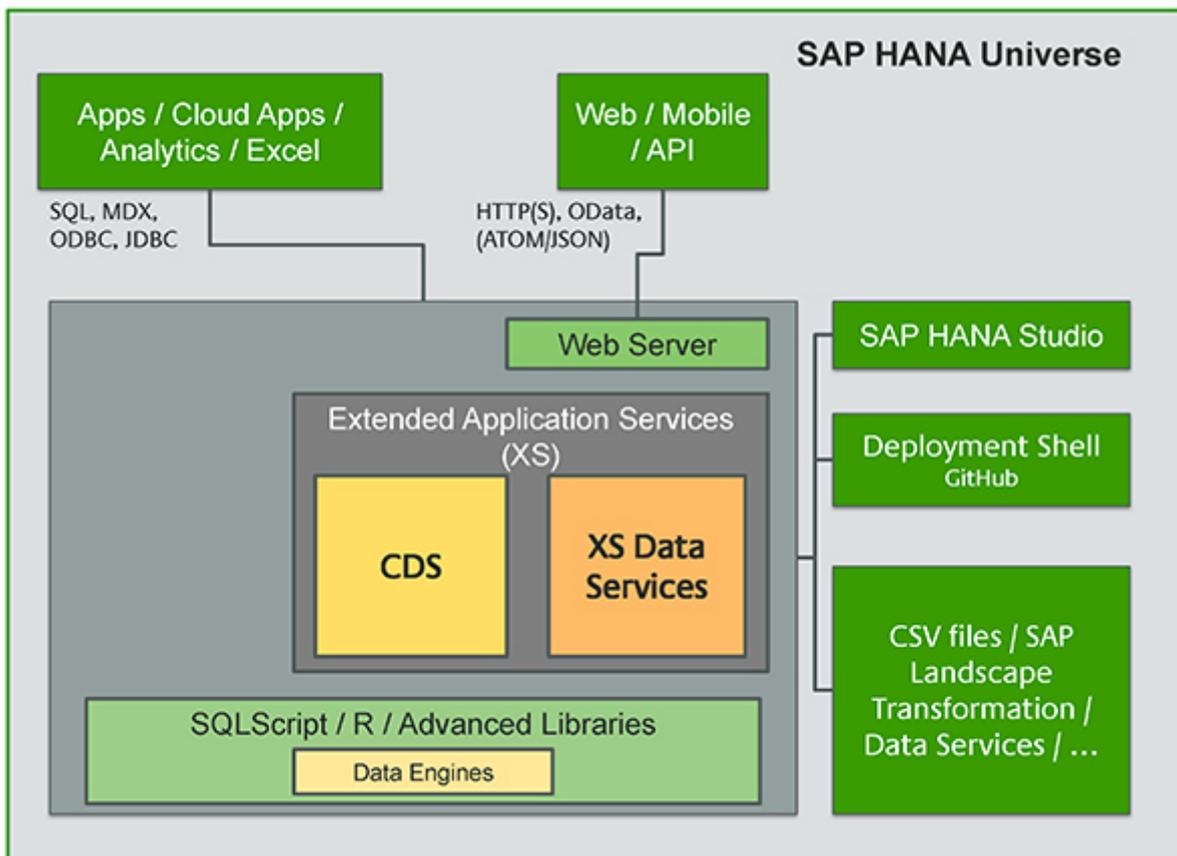


Abbildung 1: Überblick SAP HANA Universum

## Entwicklungsumgebungen

Zur Entwicklung von Software mit SAP HANA werden insgesamt zwei Entwicklungsumgebungen bereitgestellt. Einerseits das Eclipse-basierte HANA Studio und andererseits die webbasierte Development Workbench. Letztere bietet nur einen Teil der im SAP HANA Studio gebotenen Features, wird von SAP aber als die zukünftige Entwicklungsumgebung angepriesen. Daher erfolgt nach und nach die Integration einiger Web Tools aus der parallel laufenden SAP Web IDE, da beide webbasierte Entwicklungsumgebungen in eine zusammengefasst werden sollen. Sofern der webbasierte Editor bevorzugt aber die Funktionsvielfalt des HANA Studios benötigt wird, so können die webbasierten Editoren seit dem SPS09 im HANA Studio aufgerufen und verwendet werden.



## SQLScript

Bei SQLScript handelt es sich um eine Ansammlung von Erweiterungen für SQL. Neben einigen Verbesserungen belaufen sich die Erweiterungen grundlegend auf die Anweisungen CREATE TYPE, CREATE PROCEDURE und CREATE FUNCTION. Das Hauptziel von SQLScript ist die Verlagerung der datenintensiven Applikationslogik in die SAP HANA Datenbank. Somit entfällt der Transfer der zu verarbeitenden Daten von der Datenbank zum Applikationsserver, da nur noch das Ergebnis an den Applikationsserver zurückgeliefert werden muss. Weiterhin können durch die Verlagerung der Logik erst die Vorteile von SAP HANA ausgekostet werden. So werden Performanceverbesserungen durch die spaltenbasierte Verarbeitung, Query Optimierungen und die parallele Verarbeitung auf Multi-Core CPUs ermöglicht. SQLScript sollte insbesondere dann verwendet werden, wenn andere SAP HANA Modellierungsansätze (z. B. Attribute Views, Analytic Views) an ihre Grenzen stoßen.

## Core Data Services (CDS)

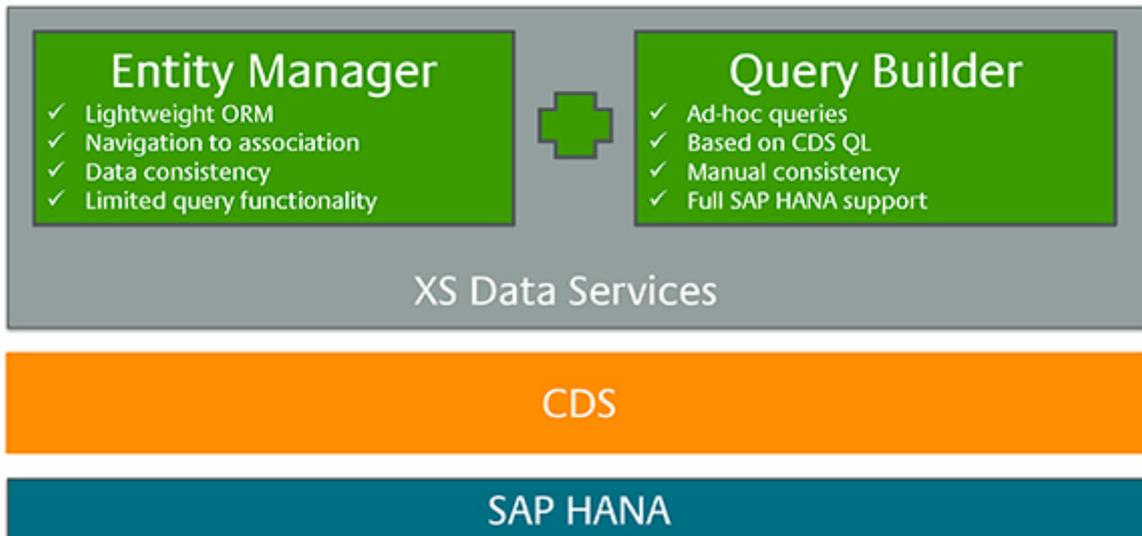
Die CDS sind Bestandteil der SAP HANA Extended Application Services (XS), einem Applikationsserver der seit SAP NetWeaver 7.4 SPS05 und SAP HANA SPS06 ausgeliefert wird.

Unter CDS wird eine Infrastruktur zur Datenmodellierung verstanden, die zur Definition und Verwendung semantischer Datenmodelle dient. Es wird als zentrales Datenmodellierungskonzept von SAP HANA verstanden. Ziel von CDS ist die Trennung von Design- und Laufzeitobjekten, so dass die Realisierung der modellierten Objekte automatisiert erfolgen kann. Zudem wird somit die Möglichkeit geschaffen, dass die definierten Objekte über beliebige Technologien konsumiert werden können. Daher erfolgt die Definition von Zielobjekten bzw. CDS Entitäten (z. B. Views, Tabellen, Beziehungen, Annotationen) mit einer CDS-eigenen Data Definition Language. Diese ist an SQL angelehnt, liegt logisch gesehen eine Sprachebene über SQL und wurde darüber hinaus um einige Features erweitert. Zudem liefert CDS eine eigene Query und Expression Language.

CDS-Skripte werden im SAP HANA Repository abgelegt und als .hdbdd-Datei gespeichert. Nach der Aktivierung eines solchen Skripts wird der entsprechende Low Level SQL-Code durch das System generiert. So kann zum Beispiel eine Tabellenstruktur per CDS definiert werden und das notwendige CREATE TABLE SQL-Statement wird automatisch generiert. Wird die Tabellenstruktur später per CDS erweitert, so wird ein passendes ALTER TABLE Statement generiert und die neue Spaltenreihenfolge berücksichtigt.

## XS Data Services (XSDS)

Bei den XS Data Services handelt es sich um eine JavaScript Library, die mit SAP HANA SPS09 als Bestandteil der SAP HANA Extended Application Services eingeführt wurde und seitdem standardmäßig installiert ist. Sie ermöglicht die Nutzung von SAP HANA Artefakten, die zuvor per CDS definiert wurden, in der XS Engine. Daher können CDS Artefakte in JavaScript als native JavaScript Objekte verwendet werden. Die Verwendung von Low-Level SQL-Statements wird somit durch High-Level Business Entitäten und deren Beziehungen abgelöst. In Abb. 2 können die Abhängigkeiten von SAP HANA, CDS und XS Data Services noch einmal im Detail betrachtet werden.



**Abbildung 2: Einordnung der XS Data Services in die SAP HANA Umgebung.**

In XSDS wurden zudem drei bisher separat geführte SAP Libraries/Projekte mit aufgenommen. Hierbei handelt es sich um den Object Relationship Mapper (XS ORM), den XS Query Builder und XS Relations. XSDS wurde als Softwarebestandteil nicht direkt in SAP HANA verankert. Somit besteht die Möglichkeit, dass XSDS jederzeit von SAP sowie dem Kunden unabhängig von SAP HANA aktualisiert werden kann.

Die XSDS JavaScript Library unterstützt den Import von CDS Entitäten und den dazugehörigen Assoziationen sowie die Manipulation dieser Objekte. Zudem werden Features wie Transaction Handling, Lazy Navigation, Custom Types sowie Views und inkrementelles Erstellen von Queries angeboten, um die Entwicklungsarbeiten weiter zu erleichtern.

## XS Java Script (XSJS)

Mit den SAP HANA Extended Application Services wird eine Reihe an Server Side JavaScript APIs ausgeliefert. Diese ermöglichen den eigenen Applikationen die Interaktion mit SAP HANA.

Durch die SAP HANA XSJS APIs werden die folgenden Interfaces bereitgestellt:

API	Beschreibung
Database	Ermöglicht den einfachen Zugriff auf die SAP HANA Datenbank mittels SQL Statements. Der \$.db Namespace zur Nutzung der API wird seit dem SAP HANA SPS09 nicht mehr empfohlen. Stattdessen wurde der neue Namespace \$.hdb eingeführt. Es sollte vermieden werden beide Namespaces in einem einzigen HTTP Request zu verwenden.
Outbound Connectivity	Ermöglicht den HTTP-Zugriff auf diverse Zielsysteme, so dass der Applikation unterschiedlichste Services angeboten werden können. Somit

	kann bspw. eine SMTP-Verbindung aufgebaut werden, über welche Multipart E-Mails verschickt werden können. Der Nutzung dieser API erfolgt über den Namespace \$.net.
Request Processing	Ermöglicht den Zugriff auf den Inhalt des aktuellen HTTP Requests, um diesen bspw. zu lesen oder einen Response zu erstellen. Zudem kann der Inhalt von einem Request oder Response manipuliert werden. Der Zugriff erfolgt über den Namespace \$.web.
Session	Bietet den Zugriff auf die aktuell verwendete SAP HANA XS Session. Somit können bspw. Zugriffsrechte oder die verwendete Sprache ausgelesen werden. Der verwendete Namespace lautet in diesem Fall \$.session.
Job Schedule	Liefert Zugriff auf das Job Scheduling Interface über welches das Definieren und Auslösen von wiederkehrenden, im Hintergrund laufenden Tasks ermöglicht wird. Zudem können neue Jobpläne hinzugefügt und gelöscht werden. Der Zugriff erfolgt über den Namespace \$.jobs.
Security	Ermöglicht über den Namespace \$.security die Nutzung der AntiVirus Engine, die sichere Speicherung im Name-Value Format und das Generieren von Hashes.
Trace	Ermöglicht den Zugriff auf alle Trace Level die zur Generierung und zum Logging der Applikationsaktivitäten genutzt werden können. Die Trace Dateien können im Reiter Diagnosis Files des SAP HANA Studios in der Administrative Perspektive eingesehen werden. Die Nutzung dieser API erfolgt über den Namespace \$.trace.
Utilities	Stellt über den Namespace \$.util die Möglichkeit bereit, XML zu parsen und ZIP-Archive zu erstellen bzw. zu bearbeiten.
XS Data Services	Liefert Zugriff auf JavaScript Utilities. Diese ermöglichen den JavaScript Applikationen die effiziente Nutzung von Datenmodellen, die über CDS erstellt wurden (siehe vorheriges Kapitel).
XS Procedures	Liefert Zugriff auf JavaScript Utilities, so dass die JavaScript Applikationen die in SAP HANA angelegten Stored Procedures wie JavaScript-eigene Funktionen aufrufen können.



## XSODATA

Bei OData handelt es sich um ein ressourcenbasiertes Web Protokoll zur Abfrage und Aktualisierung von Daten, welches ebenfalls über die SAP HANA XS bereitgestellt wird. Die zur Verfügung stehenden Operationen werden hierbei über HTTP-Befehle wie bspw. GET, PUT, POST oder DELETE realisiert. Zudem wird durch OData die Uniform Resource Indicator (URI) Syntax zur Identifikation von Ressourcen spezifiziert. Sofern Daten transferiert werden, so erfolgt dies per HTTP unter Verwendung des Atom- oder JSON-Formats.

OData soll die Erstellung von standardbasierten Applikationen ermöglichen, die unabhängig vom verwendeten Gerät und der verwendeten Plattform sind. Hierfür wird mithilfe von OData ein abstraktes Datenmodell definiert und ein Protokoll, das den Zugriff auf Daten mit einem beliebigen Client auf eine beliebige Datenquelle ermöglicht. Als Clients können bspw. Web Browser, Mobilfunkgeräte, Business Intelligence Tools oder selbst entwickelte Applikationen zur Anwendung kommen. Als Datenquellen sind u.a. Datenbanken, Content Management Systeme, Views oder die Cloud denkbar.

Der Datenaustausch per OData involviert die folgenden Elemente:

- **OData Datenmodell**  
Dieses stellt eine allgemeine Methode zur Organisation und Beschreibung von Daten dar, ohne dass die Art der Speicherung eine Rolle spielt. Die Umsetzung erfolgt über das Entity Data Model<sup>1</sup> (EDM).
- **OData Protokoll**  
Dieses ermöglicht dem Client einen OData Service abzufragen. Neben den üblichen REST-basierten Befehlen wie create, read, update und delete kommt eine OData-definierte Query Language zur Anwendung. Die Übertragung der Daten per OData Service erfolgt entweder über ein XML-basiertes Format, das durch Atom/AtomPub definiert wird oder per JSON.
- **OData Client Libraries**  
Diese ermöglichen den Clients den Datenzugriff über das OData Protokoll.
- **OData Services**  
Der OData Service stellt die Schnittstelle für den eigentlichen Zugriff auf die in der SAP HANA Datenbank gespeicherten Daten bereit. Er nutzt somit das OData Protokoll sowie den Data Access Layer, um die Daten aus dem datenbankeigenen Format in ein für den Client lesbares Format zu konvertieren.

## SAP River

Bei SAP River handelt es sich um eine Entwicklungsumgebung die auf der SAP HANA Plattform basiert. SAP River beinhaltet die SAP River Definition Language (RDL), welche zur Entwicklung von nativen Backend-Anwendungen für SAP HANA gedacht war. Dem Entwickler wurde mit SAP River die Möglichkeit geboten, alle semantischen Elemente einer Applikation, einschließlich Datenmodell, Data Constraints, Business Logik und rollenbasierte Zugriffsberechtigungen innerhalb einer einzigen integrierten Programmspezifikation zu designen.

---

<sup>1</sup> Reihe von Konzepten, die die Datenstruktur unabhängig von der gespeicherten Form beschreiben.



SAP River wurde jedoch mit dem SAP HANA SPS09 aufgrund diverser Gründe eingestellt und steht somit nicht mehr als Standalone Entwicklungsumgebung in SAP HANA zur Verfügung. Die Bestandteile von SAP River wurden jedoch wiederverwertet. Der SAP River Application Explorer wurde als SAP OData Explorer als allgemeines OData Test- und Datengenerierungs-Werkzeug mit XSODATA Services Unterstützung im SAP HANA SPS09 aufgenommen. Die restlichen SAP River Konzepte sind in angepasster Fassung in die XSDS eingeflossen.

## SAP HANA REST APIs

Die SAP HANA REST API basiert auf dem Orion<sup>2</sup> Server und dessen Client APIs und erweitert diese um zusätzliche Features. Die SAP HANA REST API unterstützt das Orion Protokoll 1.0, so dass mit Entwicklungswerkzeugen direkt auf das SAP HANA Repository zugegriffen werden kann. Somit wird in Hinsicht auf die Nutzung des SAP HANA Repositories einerseits die Arbeit mit SAP HANA Tools erleichtert und andererseits die Nutzung von Orion-basierten externen Tools ermöglicht.

Für die SAP Tools wurde das Orion Server Protokoll um die folgenden SAP HANA spezifischen Features erweitert:

- Aktivierung von Design-Time Artefakten im Repository
- Ausführung von Change-Tracking Operationen
- Durchsuchen des Datenbankkatalogs

Durch die SAP HANA REST API werden die folgenden Interfaces bereitgestellt:

API	Beschreibung
File	Ermöglicht die Navigation durch Dateien und Verzeichnisse sowie das Bearbeiten von Dateien und Verzeichnissen per HTTP.
Workspace	Erstellen und Bearbeiten von Workspaces und Projekten per HTTP. Das Workspace- und Projekt-Konzept wurde unverändert von Eclipse übernommen.
Transfer	Importieren und exportieren von Packages und Dateien.
Metadata	Ermöglicht Such- und Autovervollständigungsszenarien durch das Abrufen von Metadaten. Aktuell können Metadaten für Tabellen, Views, Prozeduren, Funktionen, Sequenzen, Schemata abgerufen werden sowie CDS Metadaten.
Change Tracking	Ermöglicht per HTTP die Nutzung von spezifischen Lifecycle-Management-Features die im SAP HANA Repository enthalten sind.
OData Services	Ein Set von OData Services, die den Zugriff auf

---

<sup>2</sup> Open Source Projekt, das unter dem Hauptprojekt Eclipse geführt wird. Ziel ist die Erstellung einer browserbasierten Open Tool Integration Plattform, die auf das Entwickeln von Webapplikationen im Web ausgerichtet ist.



	Konfigurationseinstellungen, die in der SAP HANA Environment gespeichert sind, ermöglichen.
Info	Liefert Informationen über die aktuelle Version der SAP HANA REST API.

## SAPUI5

SAPUI5 ist eine User Interface Technologie, die auf JavaScript, CSS und HTML5 basiert. Demzufolge ist SAPUI5 Bestandteil von SAP Fiori, den neuen SAP User Experience Guidelines. Die mit SAPUI5 erstellten Applikationen werden aufgrund der verwendeten Technologien über einen Webbrowser aufgerufen und in diesem wie eine Website verwendet. Eine Open Source Variante von SAPUI5 wird von SAP als OpenUI5 unter <http://openui5.org> angeboten. Da SAP der OpenAjax<sup>3</sup> Allianz angehört, ist die Kompatibilität zu anderen JavaScript Libraries gegeben. Die Speicherung der SAPUI5-Libraries sowie der SAPUI5-Applikationen kann bspw. auf einem SAP NetWeaver Application Server oder in der SAP HANA Cloud Platform erfolgen. Der Datenzugriff von einer Applikation auf die in SAP HANA gespeicherten Geschäftsdaten sollte idealerweise durch Verwendung des OData Modells über einen SAP Gateway erfolgen. Wie andere Ajax-basierte Applikationen ist auch SAPUI5 auf dem aktuellen Stand der heutigen Entwicklungen und bietet Features wie Data Binding Types, Translation Support, Keyboard Interactions, Barrierefreiheit, vordefinierte Layouts für Standard Anwendungsfälle, Responsiveness und vieles mehr.

## XS Administration Tool

Das SAP HANA XS Administration Tool gehört ebenfalls zum Bereich der Extended Application Services. Es handelt sich hierbei um ein webbasiertes Werkzeug zur Administration der grundlegend relevanten Themen, die den Applikationsentwicklungsprozess sowie die XS Umgebung betreffen. Die folgenden Bereiche können nach Belieben administriert werden:

- XS Artifact Administration
- SAML Service Provider
- SAML Identity Provider
- SMTP Configuration
- Trust Manager
- XS Job Dashboard

## SAP HANA Test Tools

Zur Unterstützung der Entwickler werden von SAP die folgenden Tools zur Unterstützung und Durchführung von Tests bereitgestellt:

- XJUnit Test Framework and Test Runner (unit.jasminex)
- XJUnit Test Coverage for XS JavaScript (unit.codecoverage)
- Test Isolation Framework (mockstar)

---

<sup>3</sup> Die OpenAjax Allianz ist eine Organisation die sich aus führenden Anbietern, Open Source Projekten und Unternehmen zusammensetzen die Ajax nutzen. Ziel dieser Organisation ist die Schaffung der Interoperabilität diverser Ajax Technologien sowie das Voranbringen der eigentlichen Ajax Technologie.



- Test Helper (unit.util)
- Developer Documentation (doc)
- Demos (demo)

Das Mock Framework wird zwar mit SAP HANA ausgeliefert, ist standardmäßig aber nicht installiert. Das Unit Test Framework ermöglicht das Testen von XS JavaScript, SQLScript sowie Views und basiert auf einer Open Source Library, die um einige Elemente erweitert wurde. Die Entwicklung des Test Codes und der Testdaten erfolgt parallel zum Produktivcode in derselben SAP HANA Instanz. Die Implementierung der Tests erfolgt in XS JavaScript in SAP HANA Studio oder in der webbasierten Development Workbench.



## Kontaktieren Sie uns //

mayato GmbH  
Am Borsigturm 9  
13507 Berlin

[info@mayato.com](mailto:info@mayato.com)

+49 / 30 4174 4270 0