

SAP HANA als mobile Anwendungsplattform

Viele Geschäftsanwendungen müssen auf Stammdaten zugreifen, die in SAP-Systemen verwaltet werden. Gleichzeitig müssen diese Daten einer Vielzahl an unterschiedlichen, auch mobilen, Clients zur Verfügung gestellt werden. Der folgende Artikel soll zeigen, wie ein solches System mit SAP HANA umgesetzt werden kann und welche Vor- und Nachteile sich durch die Verwendung von SAP HANA ergeben.

Um das Szenario zu illustrieren, kann man sich folgendes Anwendungsbeispiel vorstellen: Es soll eine Anwendung zur Unterstützung von Service-Mitarbeitern erstellt werden. Die Service-Mitarbeiter sind bei Kunden vor Ort unterwegs, um Reparaturen oder Wartungen durchzuführen, benötigen also eine Mobile-Anwendung für ihre Arbeit. Diese stellt den Service-Mitarbeitern Daten zu Geräten des Kunden bereit und hält zu jedem Gerät eine Service-Historie vor. Die Geräte-Daten werden in einem SAP-System gehalten und gepflegt. Die Service-Anwendung verwaltet die Service-Historien. Teile der Service-Daten müssen wieder in das SAP-System übertragen werden, um beispielsweise Abrechnungen zu erstellen.

Um die Anwendung umzusetzen, werden also eine mobile Anwendung als Frontend und ein Backend zur Datenhaltung und den Zugriff auf SAP-Daten benötigt. Die Kommunikation zwischen Front- und Backend erfolgt über eine REST-Schnittstelle. Für die Umsetzung gibt es diverse technologische Ansätze – im Folgenden sollen sämtliche Bestandteile anhand des von SAP angebotenen Technologiestacks realisiert werden.

Die nebenstehende Grafik beschreibt den Aufbau der skizzierten Service-Anwendung.

Technologisch gesehen verbirgt sich hinter der Applikation im Backend eine HANA-Infrastruktur mit XS-Services und ein SAP-System. Das HANA-System fungiert dabei als Master für die Bewegungsdaten, während das SAP-System der Master für die Stammdaten ist.

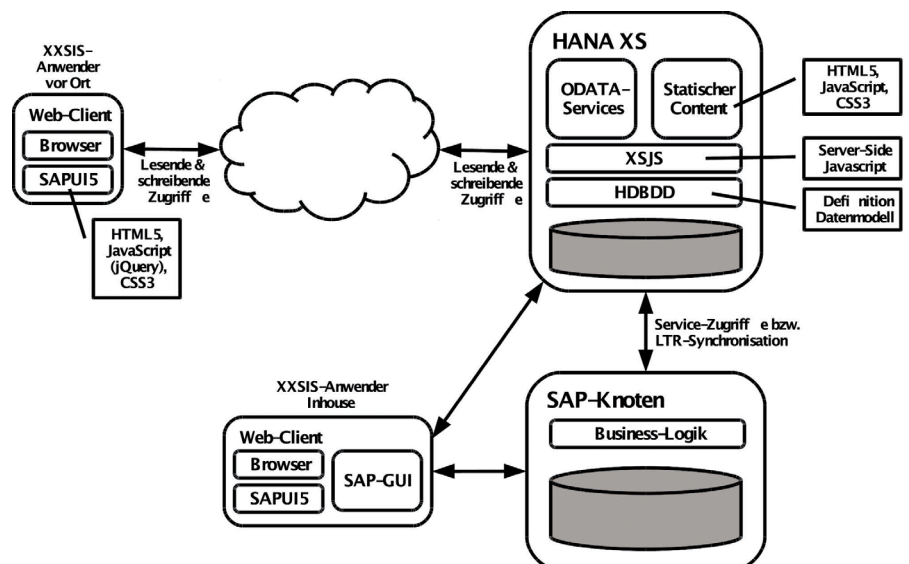
Die HANA-Infrastruktur beinhaltet neben einer reinen Datenbank auch einen Webserver, der zusätzlich zur Bereitstellung von Webcontent in der Lage ist, durch eine inte-

grierte Laufzeitumgebung (die sog. XS-Engine) serverseitig JavaScript auszuführen. Wenn nur einfache lesende oder schreibende Operationen notwendig sind, kann der Zugriff auf die Daten über sog. ODATA-Services [2] erfolgen. Diese folgen dem REST-Prinzip und sind u.a. mit JavaScript einfach zu verwenden. Der Entwickler erhält hierdurch auf vergleichsweise einfache Art und Weise den kompletten Satz an CRUD-Operationen für eine Tabelle. Komplexe Geschäftslogik kann mit der bereits erwähnten XS-Engine oder direkt in der HANA-Datenbank umgesetzt werden. Die Zugriffe auf die Services und Daten sind durch das integrierte SAP-Rechtesystem geschützt.

Die Synchronisierung der Daten zwischen HANA und SAP erfolgt mit dem sog. Landscape Transformation Server (LTR). Bei dem LTR handelt es sich im Wesentlichen um einen Dienst, der zwei Datenquellen, in diesem Fall Datenbanktabellen, anhand

von gegebenen Strategien synchron hält. So ist es beispielsweise möglich, Änderungen an den Stammdaten sofort in die zugehörigen HANA-Datenbanktabellen zu synchronisieren. Die Synchronisierung von SAP nach HANA erfolgt durch Funktionalität des LTR. Für die Synchronisierung von HANA Richtung SAP-System müssen entsprechende Services auf dem SAP-System implementiert werden, die von HANA angesteuert werden, sobald Daten über das Frontend geändert wurden. Typischerweise müssen diese Services im SAP-System in ABAP implementiert werden.

Da das primäre Einsatzszenario der Anwendung der Einsatz vor Ort ist, lag bei der Entwicklung des Frontends der Fokus auf der Verwendung von mobilen Geräten. Für diese Aufgabe bietet sich das von SAP entwickelte JavaScript-Framework SAPUI5 in Verbindung mit Apache Cordova an. Bei SAPUI5 bzw. dessen Open-Source-Variante OpenUI5[3] handelt es sich um ein von SAP



entwickeltes JavaScript-MVC-Framework, vergleichbar mit Frameworks wie das momentan sehr populäre AngularJS. SAPUI5 baut auf jQuery auf und ist dadurch relativ schnell erlernbar. Außer der reinen MVC-Struktur bietet SAPUI5 eine große Anzahl an fertigen Oberflächen-Komponenten und Funktionalitäten für den Zugriff auf Daten. Apache Cordova wird ausschließlich dafür benötigt, die SAPUI5-Anwendung in einen nativen Anwendungsrahmen zu verpacken. Im skizzierten Anwendungsszenario hat SAPUI5 den großen Vorteil, sich nahtlos in die von SAP HANA bereitgestellten ODATA-Services zu integrieren. Die bereitgestellten Komponenten sind auf die Anbindung von ODATA-Services ausgelegt. Dadurch wird das Databinding erleichtert und komplexe Suchen oder Filter sind einfach umzusetzen. Im Gegensatz zu AngularJS bringt SAPUI5 fertige Lösungen für das Layouting und Styling der Anwendung mit. Das Layouting ist responsiv, was die breite Unterstützung von unterschiedlichen mobilen Endgeräten vereinfacht. Das vorgefertigte Look and Feel hat einerseits den Vorteil, dass schnell eine gut aussehende Anwendung erstellt werden kann. Andererseits hat das den Nachteil, dass Anpassungen sich sehr aufwändig gestalten können. SAP hat diesem Problem ein Stück weit entgegengewirkt und stellt einen Theme Designer zur Verfügung, der Anpassungen am Look and Feel vereinfacht.

Ein weiterer wichtiger Aspekt bei mobilen Applikationen ist die Offlinefähigkeit. Die Applikation sollte in der Lage sein, die in der HANA-Datenbank abgelegten Daten zumindest teilweise vorzuhalten, um etwaige Verbindungsstörungen zu kompensieren oder gar einen kompletten Offlinebetrieb zu ermöglichen. SAPUI5 bietet dazu leider keine fertige Lösung. Die technische Umsetzung bedient sich zum Einen sog. „storages“, einem SAPUI5-Konstrukt, um beliebige Daten strukturiert lokal abzulegen, und zum Anderen der Möglichkeit, Callbacks an den Datenabfragen zu registrieren. Alle Zugriffe auf Daten können dadurch – falls nötig – auf die lokale Datenbasis umgeleitet werden. Bei Verfügbarkeit des Backends kann die lokale Datenbasis dann wieder mit dem Backend synchronisiert werden.

Die Hardware-Anforderungen für ein SAP-HANA System sind sehr hoch - die Minimumkonfiguration für ein System erfordert mindestens 128 GB an Hauptspeicher und

Literatur & Links

- [1] SAP HANA. [Online] <http://www.saphana.com/welcome>.
- [2] ODATA [Online] <http://www.odata.org/>
- [3] OpenUI5 [Online] <http://openui5.org/>
- [4] SAP HANA Cloud Platform [Online] <http://hcp.sap.com/index.html>
- [5] HANA auf AWS [Online] <https://aws.amazon.com/de/sap/>

einer nicht näher genannten Menge an Plattenspeicher. Unsere Installation wird momentan mit 500 GB Plattenplatz betrieben. Alternativ gibt es die Möglichkeit HANA-Instanzen in der Cloud zu nutzen, sei es über SAP[4] oder über Amazon Web Services (AWS) [5]. Während dies die nicht unerheblichen Investitionen in Hardware spart, entstehen aber Kosten je nach Größe des Systems und produzierter Rechenlast. Die Verwendung von HANA in der Cloud stellt für Entwicklungszwecke durchaus einen gangbaren Weg dar. Beim Cloud-Betrieb gibt es allerdings einige Einschränkungen in Bezug auf Datenmodellierung und Zugriffsmöglichkeiten auf die HANA-Instanz. Um diesen Einschränkungen zu entgehen, empfiehlt sich der Betrieb einer eigenen HANA-Installation. Insbesondere bei der Anbindung eines weiteren SAP-Systems zum Zwecke der Synchronisierung der Daten bietet eine eigene Plattform deutlich mehr Flexibilität.

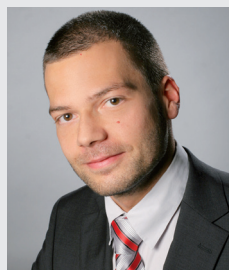
SAP bietet mit HANA einen Stack der sich grundsätzlich als Plattform für komplexe mobil-zentrierte Anwendungen eignet.

Sowohl bei Frontend- als auch Backendentwicklung bewegt man sich in der JavaScript-Welt. Die Erstellung von Services anhand von vorhandenen Datenbanktabellen gestaltet sich sehr einfach und bietet dennoch ausreichende Flexibilität, um komplexere Fälle abzubilden. Erfreulicherweise verwendet SAP bekannte Bestandteile wie z.B. jQuery. Entwickler können daher vorhandenes JavaScript-Wissen auch in der SAP-Welt gut anwenden.

Mit SAPUI5 setzt SAP auf moderne Webtechnologien und bietet mit ihrem MVC-Framework das nötige Rüstzeug, um ansprechende Webapplikationen zu erstellen. Das Framework ist auf das Zusammenspiel mit HANA abgestimmt und führt in Konstellation mit HANA schnell zu brauchbaren Ergebnissen. Ohne HANA bietet das Framework aber wenig Vorteile gegenüber anderen Alternativen.

Aufgrund der mit den Lizenzen und dem Betrieb der Hardware verbundenen Kosten eignet sich der Einsatz primär für Betreiber, die bereits in der SAP-Welt verwurzelt sind und optimalerweise schon SAP-Systeme im Einsatz haben. ||

Die Autoren



|| Dr. Patrick Schmidt
(patrick.schmidt@excellent.de)
ist Senior Software Engineer bei der eXXcellent solutions GmbH in Ulm. Sein Schwerpunkt liegt in der Konzipierung und Implementierung von Web- & Rich-Client-Anwendungen und der Entwicklung von HANA-Applikationen.



|| Dr. Ralph Guderlei
(ralph.guderlei@excellent.de)
ist Technology Advisor bei der eXXcellent solutions GmbH in Ulm. Neben der Arbeit als Architekt/Projektleiter in unterschiedlichen Kundenprojekten berät er Teams in technologischen und methodischen Fragestellungen. Seine Erfahrungen gibt er gerne auf Konferenzen und in Fachartikeln weiter.