Data Vault als Grundlage für DevOps

Data Warehouse Automation

Swiss Re, eines der weltweit größten Rückversicherungsunternehmen, hat sein Data Warehouse für Risikomanagement ersetzt. Vorgaben waren dabei Zukunftssicherheit, ein hoher Automatisierungsgrad, Kompatibilität mit einem agilen Entwicklungsansatz und strenge Anforderungen an die Auditierbarkeit. Die Anforderungen wurden durch eine Metadaten-getriebene Generierung eines Data Vault DWH umgesetzt. Dadurch wurde die benötigte Zeit, um neue Datenquellen anzubinden, massiv reduziert und gleichzeitig die Wartbarkeit verbessert.

Als wir bei Swiss Re, einem der weltweit größten Rückversicherungsunternehmen, entschieden, unser existierendes Data Warehouse für Risikomanagement zu ersetzen, bestand der Wunsch nach einer zukunftssicheren Lösung, die sowohl unsere strengen Anforderungen an Auditierbarkeit erfüllt als auch unseren agilen Entwicklungsprozess optimal unterstützt. Besonderes Augenmerk lag auf der konsequenten Umsetzung der DevOps-Philosophie und dem Wunsch, alle repetitiven Arbeitsschritte bei Entwicklung, Testen und Deployment des Data Warehouse zu automatisieren.

Wir fanden mit Data Vault eine Architektur, die unsere Anforderungen bestens bedient, und mit der 2150 GmbH einen kompetenten Partner mit langjähriger Erfahrung und einem umfangreichen, ausgereiften Toolset, dem „Datavault Builder Framework“. Im Rahmen des Projekts wurde das Framework so erweitert, dass das Data Warehouse vollständig über ein Swiss-Re-spezifisches Metamodell definiert und generiert werden kann. Als Datenbank kam Oracle 12c auf Exadata zum Einsatz. Als ETL-Tool für das Staging der Daten wurde Informatica PowerCenter verwendet.

Paradigmenwechsel

Data Vault bringt einen Paradigmenwechsel im Data Warehousing mit sich: Daten werden zuerst so geschrieben und historisiert, wie sie die Quelle liefert, weil dies der Realität entspricht. Erst in einem zweiten Schritt werden Verarbeitungsregeln angewendet. Dies kommt einerseits der agilen Entwicklung entgegen, weil man dem Anwender sehr schnell Zugang zu den Rohdaten geben kann, andererseits lässt sich dieser erste Teil praktisch komplett automatisieren. Der zweite Teil der Definition der Business-Regeln kann dann in mehreren Zyklen nach und nach umgesetzt werden. Nach jedem Zyklus stehen dem Business-Anwender die Resultate zur Verfügung und er muss nicht im Blindflug neue Regeln definieren (siehe Abbildung 1).

Anbindung neuer Datenquellen

Der erste Schritt der Automatisierung, das Anbinden einer neuen Quelle, umfasst die Erstellung der Staging-Tabellen, Staging-Views und Data-Vault-Artefakte. Datenanalysten der Swiss Re erstellen Schnittstellendefinitionen zu den Quellsystemen in einem definierten Format. Diese Definition wurde durch wenige Spalten erweitert, in denen Metadaten über den Inhalt abgelegt werden, zum Beispiel was der Business-Schlüssel (BK) und was allfällige Fremdschlüssel sind (FK). Basierend auf diesem Inhalt wird der richtige Hub erstellt, aus dem BK wird der Hub-Load und aus den FKs die Links generiert, alle anderen Attribute werden automatisch in Satelliten aufgenommen.

Für die Übertragung aus der Quelle zum Data Vault hatte Swiss Re bereits einen Generator für ETL Staging Loads mit Informatica im Einsatz, der auch weiterhin genutzt wird. Diese Loads beinhalten keine Logik, die Daten verändert,

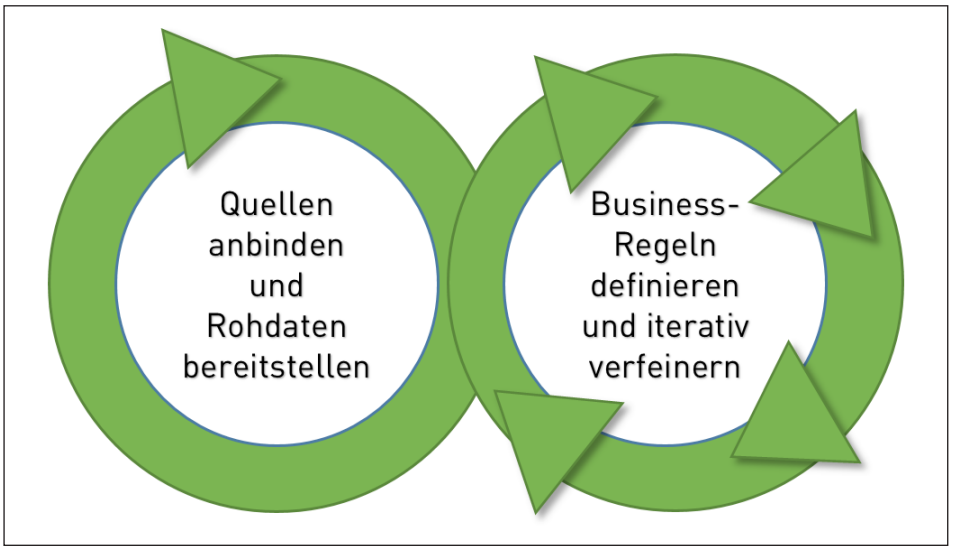

Abb. 1: Data-Vault-Vorgehensweise bei der Anbindung neuer Datenquellen

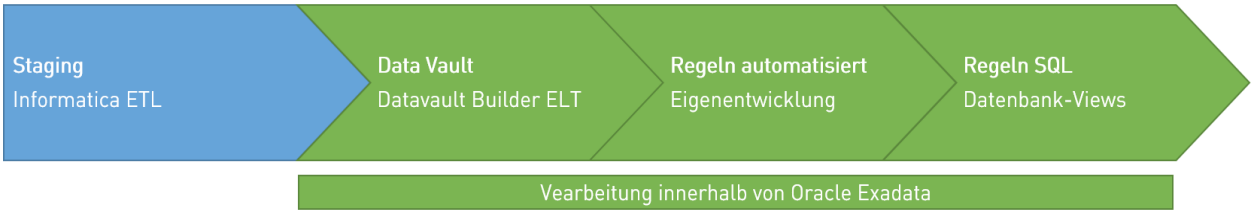

Abb. 2: Datenverarbeitung von der Quelle bis zum Ziel

sondern dienen lediglich dem Transfer von Quell- zu Zielsystem. Dieser Generator arbeitet mit den gleichen Metadaten wie der Datavault Builder. Somit kann durch einmalige Definition der Metadaten der gesamte Anbindungsprozess automatisiert werden.

Das Datavault Builder Framework übernimmt anschließend die Erstellung der nötigen Datenbankobjekte wie Hubs, Links und Satelliten und erstellt innerhalb der Datenbank die nötigen ELT-Loads mit Historisierung (SCD Typ 2). Dabei werden alle Standard-Artefakte inklusive Tracking-Satelliten automatisch erstellt, die um Swiss-Re-spezifische Objekte ergänzt wurden. Um die erhöhten Anforderungen an die Auditierbarkeit der Datenloads zu adressieren, wurde im Framework konfiguriert, dass bei der Erstellung von Satelliten und Links automatisch ein spezieller Tracking-Satellit erzeugt wird. Dieser Tracking-Satellit speichert, welche Schlüssel in welchen Loads vorhanden waren. Die Beladung dieser Tracking-Satelliten erfolgt wiederum mit den Standard-Methoden des Frameworks (Abbildung 2).

Auslesen des Data Vault

Bei der Abfrage der Daten haben die Konsumenten unterschiedliche Anforderungen. Ein System möchte die Quelldaten pro Datenlieferung abfragen können, ein anderes möchte per Stichtag die aktuellsten Informationen sehen, und andere Abnehmer wiederum möchten eine SCD-Typ-1-Sicht (As-of-Now-Sicht). Um die Abfragen zu vereinfachen, wurden deshalb Abfragesatelliten eingeführt. Diese liefern zum Teil virtuell (Views), teilweise aber auch materialisiert die verschiedenen Sichten auf die Daten. Die Erstellung dieser Abfragesatelliten wird ebenfalls direkt vom Framework angestoßen. Somit stehen viele Abfragen von unverknüpften Daten schon direkt nach der Generierung der

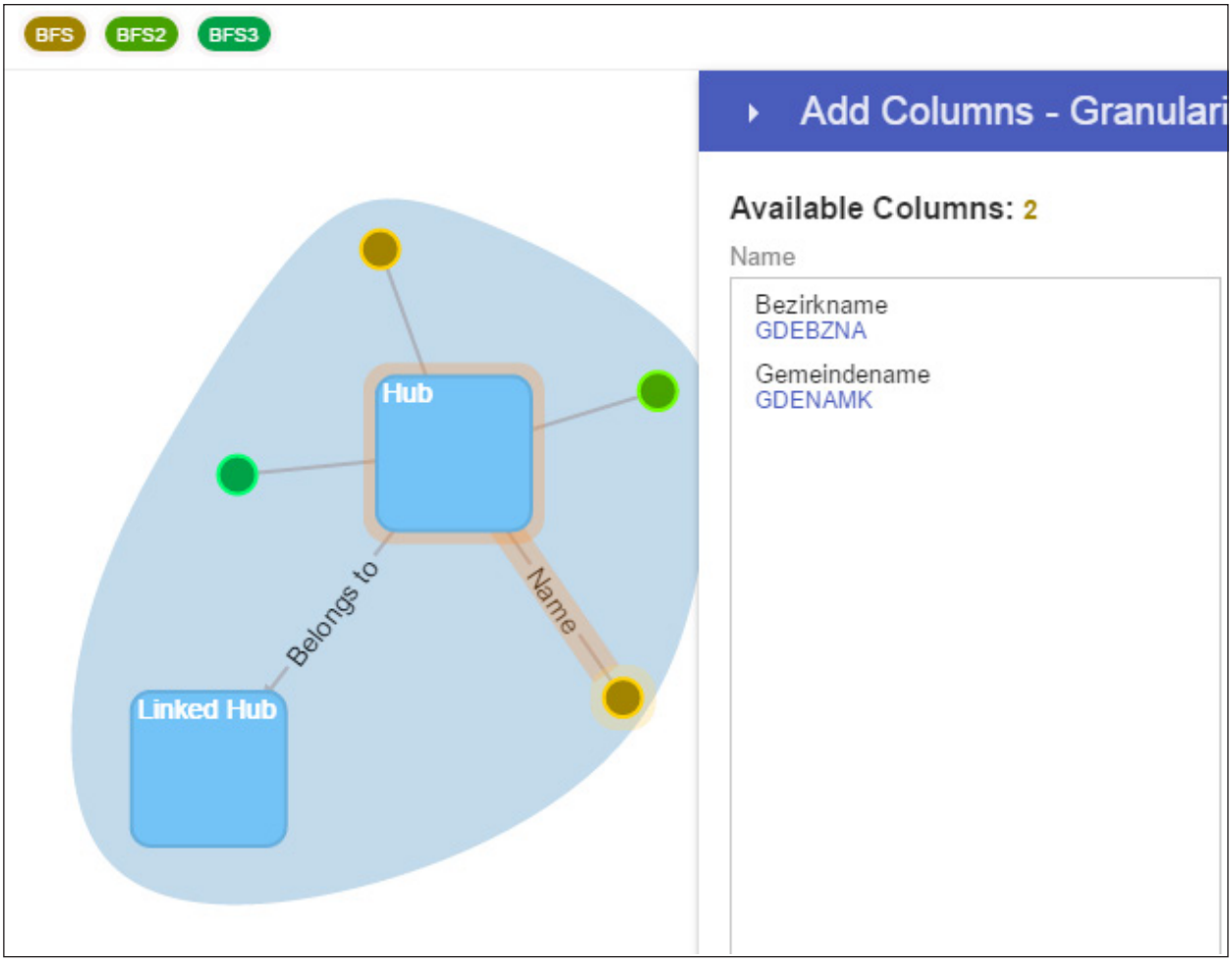

Abb. 3: Automatisierte Denormalisierung der Data-Vault-Objekte (Symbolbild aus „Datavault Builder“)

Data-Vault-Objekte zur Verfügung. Die Durchlaufzeit vom Empfang der Metadaten bis zur Bereitstellung der Daten in dieser ersten Version an den Abnehmer dauert somit wenige Minuten anstatt wie früher mehrere Tage.

Dies bedeutet nicht, dass alle Datenkonsumenten bereits eine finale Lösung erhalten, aber der erste Schritt kann innerhalb eines Tages umgesetzt werden und der Konsument erhält einen ersten Blick auf die Daten. Zudem gibt es Systeme, die effektiv mit historisierten, aber unveränderten Daten weiterarbeiten. Diese sind mit der automatischen Anbindung bereits komplett bedient. Die anderen Konsumenten können mit Sicht auf die Daten einfacher definieren, welche weiteren Verarbeitungsschritte sie gerne in Auftrag geben möchte. Zudem kann die Implementierung bereits auf Testsysteme und die produktive Umgebung ausgerollt werden und somit eine Historie der Daten aufgebaut werden, welche die Analyse ebenfalls unterstützt.

Auf diese Weise stellt das Data Warehouse keine Engstelle im Entwicklungsprozess dar, sondern unterstützt aktiv die agile Entwicklung von konsumierenden Systemen und Reports.

Business Ruleset

```
1  CREATE VIEW business_rules.hub_s_bfs
2    AS
3
4  SELECT
5
6    bo."Business Key for Hub Hub",
7    bo."Bezirkname",                -- "GDEBZNA" from satellite Name of hub Hub in system BFS
8    bo."GDEKT"                      -- "GDEKT" from default satellite of hub Hub in system BFS2
9
10 FROM businesslayer.hub_s_bfs AS bo
11 where "Bezirkname" = 'Zürich'     -- manually added business rule
12 ;
```

Abb. 4: SQL-Layer für Business-Regeln, der auf dem denormalisierten Objekt basiert

Einbau von Business-Regeln

Der Einbau von Business-Regeln wurde in zwei Kategorien aufgeteilt: in automatisierbare und nicht automatisierbare Teile.

Automatisiert wurden dabei Regeln, die sehr häufig vorkommen und einfach zu beschreiben sind, wie das Denormalisieren der Objekte aus dem Core, Umbenennungen von Attributen, Multiplikation mit Faktoren und die Typenkonversion. Dabei werden Regeln auf Attributebene direkt bei der Erhebung der Metadaten hinterlegt. Die Definition der Denormalisierung erfolgt über ein eigenentwickeltes Java-GUI, das die Metadaten der Datenbank ausliest und dem Anwender alle verfügbaren Attribute ausgehend von einem Hub, der die Granularität definiert, in einer Baumhierarchie zur Auswahl stellt. Dieses GUI erzeugt XML-Dateien, die dann wiederum von der Datenbank in Views umgesetzt werden. Hier wurde auf eine Eigenentwicklung gesetzt, weil das entsprechende Modul des Datavault Builder zum Implementationszeit noch nicht zur Verfügung stand (Abbildung 3).

Weil diese Regeln nicht alle Anforderungen abdecken, ist eine weitere virtuelle Schicht auf den automatisch generierten Views vorhanden, in der die volle Funktionalität von SQL angewendet werden kann. Da aber keine Join-Operationen anfallen und die Felder bereits richtig benannt sind, ist die hier definierte Logik sehr übersichtlich (vgl. Abbildung 4).

Durch die Virtualisierung der Regelschichten sind Änderungen sehr schnell umsetzbar. Die Virtualisierung der Abfrageschicht hat aber auch ihre Grenzen. Werden zu viele Regeln aufbauend hintereinandergeschaltet, müssen gewisse Zwischenschichten materialisiert werden. Dies war auch der Grund, wieso am Ende einige virtuelle Abfragesatelliten im Core auf eine materialisierte Version umgestellt wurden.

Fazit

Durch die Automatisierung dauert die Anbindung neuer Quellen nach der Definition durch die Datenanalysten nur wenige Minuten, und dies mit einer reproduzierbar hohen Qualität. Der Data-Vault-Ansatz passt perfekt zur agilen Vorgehensweise und erlaubt die Erstellung neuer Objekte und das Deployment in kleinen Schritten. Dies bildet auch die Grundlage für die nächsten Schritte in Richtung DevOps.

Der Einsatz eines Metadaten-getriebenen Ansatzes mit einem spezialisierten Werkzeug wie dem Datavault Builder ist zu empfehlen, weil im Vergleich zu einer Eigenentwicklung damit ein Projekt massiv beschleunigt werden kann und damit auch der Aufwand reduziert wird. Bei aller Automatisierung werden aber weder Datenanalysten noch Entwickler in Zukunft arbeitslos: die Definition der Metadaten erfolgt weiterhin durch Menschen, und bei der automatischen Erstellung komplexer Business-Regeln aus normaler Sprache ist der gordische Knoten noch nicht durchtrennt worden.

Der Ansatz hat sich in der Praxis bewährt und das Data Warehouse wird schrittweise um neue Funktionalitäten ergänzt. Neben kontinuierlichen Verbesserungen der Automatisierung, insbesondere beim Deployment der Datenbankobjekte, sind außerdem Erweiterungen für nicht relationale Datenquellen und der Einsatz eines Datenvirtualisierungs-Tools für den Business Layer des Data Vault geplant.

Uwe Raetz ist bei Swiss Re für die IT-Architektur und die Qualitätssicherung im Bereich Risikomanagement verantwortlich. Er hat langjährige Erfahrung mit agiler Entwicklung und DevOps im Data Warehousing.
E-Mail: UweMarcus_Raetz@swissre.com

Petr Beles ist Data-Vault-Spezialist und hat in den vergangenen vier Jahren den Aufbau mehrerer Data-Vault-DWHs betreut. Als Product Owner betreut er bei der 2150 GmbH die Entwicklung des „Datavault Builder".
E-Mail: petr.beles@2150.ch