➢ Pressebericht vom 9. Oktober 2014

Big Data durch SEPA: Mehr Zahlungen, mehr Transaktionen, mehr Kosten

Mit der richtigen Technologie Kosten senken und Akzeptanz erhöhen

Massenhaft Zahlungen – Big Data im Zahlungsverkehr

SEPA - die Einführung eines einheitlichen Zahlungsverkehrs in Europa ist eine große Herausforderung für die Verbraucher. Es bedeutet die Eingabe langer Zahlenreihen für eine Überweisung, statt Kontonummer und Bankleitzahl. Dementsprechend gering die Akzeptanz und ein wichtiger Grund dafür, dass die Einführung zeitlich verschoben worden ist.

Allerdings stecken noch mehr Herausforderungen in diesem großen europäischen Projekt: Die Menge der Zahlungsverkehrs-Transaktionen wird weiter zunehmen. Manche Anforderungen aus dem SEPA-Projekt machen die Speicherung jeder einzelnen Transaktion notwendig. Das bleibt so, auch wenn die SEPA-Umstellungsphase verlängert worden ist. Und das ist ein Thema für die IT-Entscheider. Hierin besteht die wahre Herausforderung der SEPA-Einführung.

Wir sprechen also im Folgenden von den großen Datenmengen, die anfallen und die unbedingt beherrscht werden müssen. Anderenfalls wird die Datenverarbeitung zum Engpass und das große Projekt bekommt an den Stellen Probleme, die der Verbraucher nicht sieht und deshalb die Auswirkungen nicht verstehen kann. Das wird der Akzeptanz nicht helfen.

Big Data - neue Lösungen in Sicht?

Für lange Zeit gab es für einen schnellen Zugriff auf Daten nur eine Speicherlösung: Relationale Datenbanken. Um alle geforderten Funktionalitäten der SEPA-Einführung abbilden zu können, müssen neue Wege beschritten werden, weil die Datenmengen derart zunehmen werden, dass schon bald relationale Datenbanken allein nicht mehr ausreichen.

Wozu Transaktionen speichern?

Wenn Gutschriften und Lastschriften ausgeführt werden und alles nach Plan läuft, passiert Folgendes: Die Transaktion wird von der Auftraggeber-Bank an die Empfänger-Bank geschickt und das Geld entweder dem Konto des Empfängers gutgeschrieben oder abgebucht.

Es kann aber auch vorkommen, dass zum Beispiel das Empfänger-Konto nicht mehr existiert oder nicht über genügend Deckung verfügt. Dann kommen die sogenannten R-Transaktionen ins Spiel: Damit wird die ursprüngliche Transaktion rückgängig gemacht. Damit das nicht mehrfach passiert, muss man erkennen können, dass eine Transaktion bereits rückgängig gemacht wurde. Dies kann man nur an Hand des gesamten Geschäftsvorfalls erkennen. Deshalb muss eine Transaktion mit einer potentiellen R-Transaktion in irgendeiner Form in Verbindung gebracht werden. Für diese Recherche muss ein Datenbestand aller Transaktionen und R-Transaktionen aufgebaut werden.

Wie speichert man Transaktionen?

Bei dem Stichwort Verbindung oder auch Beziehung kommt einem Informatiker erst einmal eine relationale Datenbank zur Ablage der Daten in den Sinn. Jede Transaktion hat potentiell eine R-Transaktion zur Folge und das muss datenbanktechnisch vorgesehen werden. Allerdings ist genau da der Haken: Nicht für jede Transaktion gibt es auch eine R-Transaktion! Man kann sogar davon ausgehen, dass mehr als die Hälfte aller Transaktionen ohne Probleme ausgeführt werden. Dann sind mehr als die Hälfte aller Fremdschlüssel – das ist die Verknüpfung zwischen der Original-Transaktion und ihrer R-Transaktion - leer. Man kann natürlich auch ohne Fremdschlüssel arbeiten. Aber worin läge dann der Nutzen in einer teuren, relationalen Datenbank?

Eine andere mögliche Technik für eine solche Anforderung ist eine In-Memory-Datenbank. So eine Datenbank ist sehr schnell, da alle Daten im Hauptspeicher des Rechners verwaltet werden. Auch diese wäre aus den oben genannten Gründen die falsche Alternative: Warum sollte man Millionen von Transaktionen im Speicher vorhalten, wenn nur wenige Prozent für die geforderten Funktionalitäten benötigt werden? Das bedeutet nur unnötig hohe Hardware-Kosten.

Damit landet man bei der Suche nach Alternativen schnell beim Thema NoSQL. Diesem sind verschiedene Konzepte für Datenbanken zugeordnet und es bedeutet nicht, dass man die Abfragesprache SQL nicht verwenden kann! Sondern Teile der Konzepte der relationalen Datenbanken, die SQL groß gemacht haben, werden durch neue Ansätze ersetzt oder ergänzt.

Die unterschiedlichen Ansätze der NoSQL-Datenbanken sind inzwischen zwar weit verbreitet, aber bei Banken offenbar wegen der vermeintlich fehlenden Transaktionssicherheit verpönt. Dabei ist Transaktionssicherheit für eine Recherche nicht notwendig: Unter Umständen sind nicht alle Daten gleichzeitig sichtbar, aber dass sie sichtbar werden, ist sichergestellt.

Viele solcher NoSQL-Datenbanken stellen keine durch die Datenbank unterstützten Beziehungen zwischen unterschiedlichen Datensätzen zur Verfügung. Wenn die Daten also keine technische Verknüpfung durch einen Fremdschlüssel in der Datenbank erhalten, haben sie erstmal nichts mehr miteinander zu tun. Erst durch die Interpretation der Daten bei einer Suche erhalten sie eine Verbindung.

Auf die Organisation kommt es an

So findet man bei einer geschickten Organisation der Informationen dann auch alle Daten, die in Beziehung stehen, ohne einen Fremdschlüssel wie in einer relationalen Datenbank abzulegen. Dies ist insbesondere für das Speichern einer R-Transaktion interessant: Statt zunächst die Original-Transaktion zu suchen und deren Primärschlüssel zusammen mit der R-Transaktion abzuspeichern, wird die Transaktions-ID der Original-Transaktion, die in jeder R-Transaktion vorhanden ist, im gleichen Feld abgespeichert - ohne Suche der Original-Transaktion! Das spart Zeit beim Speichern von R-Transaktionen. Bei der Suche kann dann mit Hilfe der Informationen über die Original-Transaktion (die Transaktions-ID, das Settlement-Datum und die Absender- und Empfänger-BIC) der gesamte Geschäftsvorfall zu der Original-Transaktion gefunden werden, ohne dass man bei der Suche je nach (R-)Transaktions-Art unterschiedliche Felder betrachtet und somit mehrere Suchanfragen ausführt, um das gewünschte Suchergebnis zu erhalten. Damit erhält man bei einer Suche für eine Reklamation sehr schnell einen Überblick darüber, was mit der Transaktion passiert ist.
Auf diese Weise schränken die oben genannten Informationen aus einer Transaktion, die sowohl in Original- als auch in R-Transaktionen immer enthalten sind, diese bereits

ein und helfen bei der Interpretation von Suchergebnissen. Damit können zwischen ihnen Beziehungen ermittelt werden. Das alles ohne technische Hilfsmittel wie Fremdschlüssel, die das Speichern von Daten sogar verlangsamen.

Support

Hinter den großen relationalen Datenbank-Systemen stecken in der Regel Firmen, die auch Support für ihre Systeme verkaufen. Genauso haben es sich auch um die NoSQL-Datenbanken Unternehmen zur Aufgabe gemacht, Support für die jeweilige Datenbank anzubieten. Diese Unternehmen wurden in der Regel von den Erfindern der jeweiligen Datenbank gegründet. Der Unterschied zu den relationalen Datenbanken ist der als Open Source zur Verfügung stehende Quellcode. Oft ist damit das jeweilige Datenbanksystem kostenlos.

Mit der richtigen Technologie Geld sparen

Die Idee der losen Kopplung von manchen NoSQL-Datenbanken sollte für ein Umdenken sorgen: Man kann zwar alles mit einer teuren relationalen Datenbank erschlagen, aber wenn man sich auch mit alternativen Konzepten beschäftigt, stellt man fest, dass diese zu erheblichen Einsparungen führen können. Und das insbesondere auch im Banken-Umfeld, wo Daten langfristig aufbewahrt werden müssen. Genau diese Datenmengen durchsuchbar zu machen, ist eine Stärke von NoSQL-Datenbanken. Dass es vielleicht auch mit relationalen Datenbanken funktioniert, ist unbestritten. Aber es gibt eben auch Nachteile, die Zeit und Geld kosten. Und beides sind bekanntermaßen knappe Güter!

Der Erfolg und die Verbraucher-Akzeptanz der SEPA-Einführung hängen von vielen Faktoren ab. Die IT-Entscheider können einen erheblichen Beitrag dazu leisten, wenn sie neue technologische Wege beschreiten.

Ansprechpartner:
Jens Kötterheinrich
Informatiker
Beckmann & Partner CONSULT
Telefon: 0521 2997320
info@beckmann-partner.de
www.beckmann-partner.de