

## Risikomanagement

# Systemarchitektur als Erfolgsfaktor !

Manuel Ammann, Urs Halter, Christian Schmid<sup>1</sup>

**Kreditrisiko-Serie (4).** Kosten, Know-How, Daten und Kreditrisiko-Steuerungsüberlegungen sind Gründe, warum ein Alleingang im modernen Kreditrisikomanagement für eine durchschnittliche Kantonal- oder Regionalbank nicht anzustreben ist. Eine Zusammenarbeit verschiedener Banken im Kreditrisikomanagement setzt nicht nur Einigkeit in bankfachlicher Sicht voraus, sondern auch eine Lösung mit einer Systemarchitektur, welche die technische Integration in bestehende Applikationslandschaften ermöglicht.

In den vergangenen Ausgaben der Schweizer Bank legten wir das Schwergewicht auf bankfachliche Aspekte eines modernen Kreditrisikomanagement-Projektes. Mehrfach haben wir dabei darauf hingewiesen, dass im Vergleich zur Messung von Marktrisiken die heute sehr dünne Datenlage bei Kreditrisikomodellen zu den zentralen Problemfeldern zu zählen ist. Gleichzeitig fehlt den meisten Banken in der Schweiz die kritische Grösse zur Durchführung eigener Steuerungsmassnahmen. Bei Verbriefungstransaktionen wird man beispielsweise auf Multi-Seller-Strukturen angewiesen sein. Hinzu kommt, dass es in vielen Banken an spezialisierten Fachkräften mit dem notwendigen ökonomischen Rüstzeug fehlt. Betrachtet man ausserdem die Kosteneinsparungen durch gemeinsame Systementwicklung, so wird klar, dass ein Alleingang für die meisten Institute eine suboptimale Lösung darstellt.

Aufgrund dieser Ausgangslage beschäftigen wir uns in diesem Beitrag mit den technischen Aspekten von Kreditrisikomanagement-Systemen. Diesen Aspekten kommt eine besondere Bedeutung zu, weil im Bereich der Kreditrisikomanagement-Systeme heute noch kaum Standardsoftware verfügbar ist. Eine Bank ist deshalb darauf angewiesen, entweder ein eigenes System aufzubauen, über eine Kooperation mit anderen Banken an einer Systementwicklung teilzunehmen oder sich einer bestehenden Lösung anzuschliessen.

Im folgenden gehen wir auf die wichtigsten Design-Ziele ein, welche in der Entwicklung des Toolsets für das Kreditrisikomanagement bei der St. Gallischen Kantonalbank und der Luzerner Kantonalbank verfolgt werden.

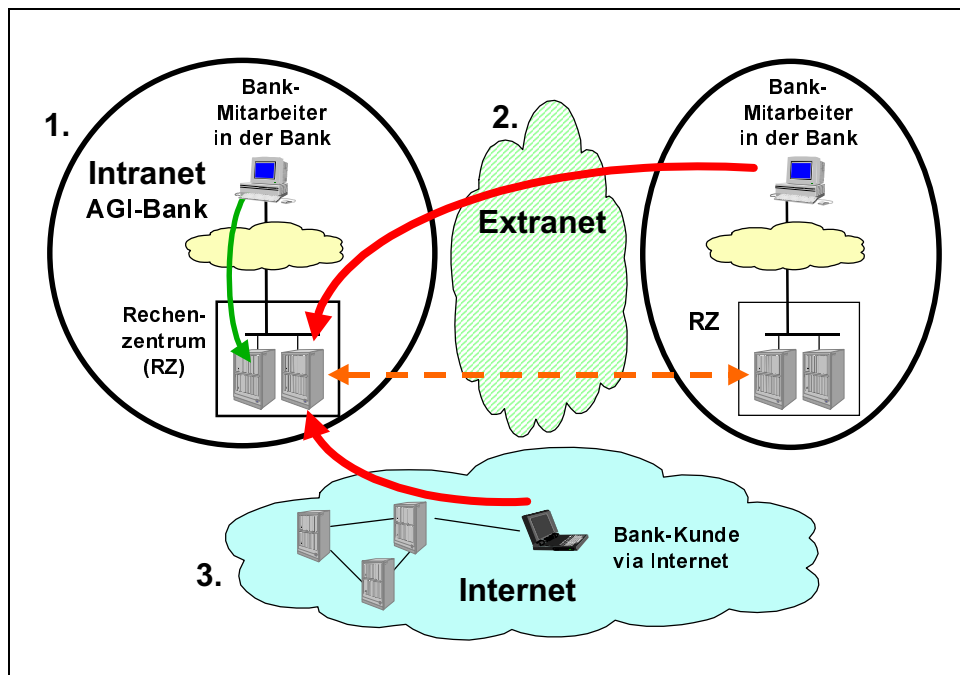
---

<sup>1</sup> Manuel Ammann ist vollamtlicher Dozent für Finanzmarkttheorie an der Universität St. Gallen, Urs Halter ist Mitglied der Direktion und Leiter Architektur bei der AGI IT Services AG. Christian Schmid ist Mitglied der Direktion und Leiter Portfoliomanagement Kredite bei der St. Gallischen Kantonalbank.



## Design-Ziel 2: Web-Fähigkeit

Wenn heute die System-Architektur für ein neues Entwicklungsprojekt mit moderner Technologie zu definieren ist, so ist sicher der Einsatz von Web-Technologie und die Verwendung von Java als Plattform in Betracht zu ziehen. Für die Entwicklung eines Kreditrisikosystems liefert Webtechnologie insbesondere den Vorteil der erleichterten Anbindung der verschiedenen Benutzergruppen an das System. Diese Anbindung ist in Grafik 2 veranschaulicht.



Grafik 2: Einbindung verschiedener Benutzergruppen

Im bankinternen Intranet-Einsatz (in der Grafik mit 1. gekennzeichnet) greift der Bank-Mitarbeiter via Browser auf die Server-Infrastruktur im Rechenzentrum zu. Zusätzlich kann die Applikation jedoch anderen Banken und deren Mitarbeitern über eine Extranet-Anbindung (2.) zur Verfügung gestellt werden und ein eigentliches Applikations-Hosting als Dienstleistung angeboten werden. Solche Dienstleistungen bedingen, dass zwischen den externen und den internen Applikationen sowie den Datenbanken ein Abgleich stattfindet, um die Konsistenz der Daten zu gewährleisten.

Schliesslich soll in Zukunft aber auch der Einsatz einzelner Module durch den Kunden via Internet (3.) erfolgen. Bei Firmenkunden ist dies z.B. die Finanzanalyse, bei Privatpersonen könnte dies auch über die Eingabe der Daten für die Ratingberechnung hinausgehen bis hin zur automatisierten Kreditvergabe via Internet.

Innerhalb der Web-Technologie besteht heute wegen verschiedenen Architektur-Modellen und unterschiedlichen Einsatz-Varianten von Java die Qual der Wahl. Entscheidend ist die Frage, ob eine Fat- oder Thin-Client Strategie verfolgt wird. Beim Fat-Client Ansatz wird die Business-Logik in Form einer Java-Applikation oder von Java-Applets auf dem Client ausgeführt, währenddem bei einer Thin-Client Lösung die Logik für Präsentationsaufbereitung und die eigentliche Business-Logik serverseitig mit dazu vorhandenen Java-Techniken implementiert wird.

Für ein Kreditrisikosystem bietet sich eine System-Architektur nach dem Thin-Client Modell mit server-seitig in Java implementierter Business-Logik und ein 3-stufiger, sogenannter 3-Tier Ansatz an. Die 3 Stufen bestehen aus einem Internet-Browser als Client für den Benutzer (Stufe 1), einem Applikations-Server für die Business-Logik (Stufe 2) und einem Datenbank-Server sowie der Integration bestehender Basis-Applikationen (Stufe 3). Gegenüber Fat-Client Ansätzen hat dieses Vorgehen folgende Vorteile:

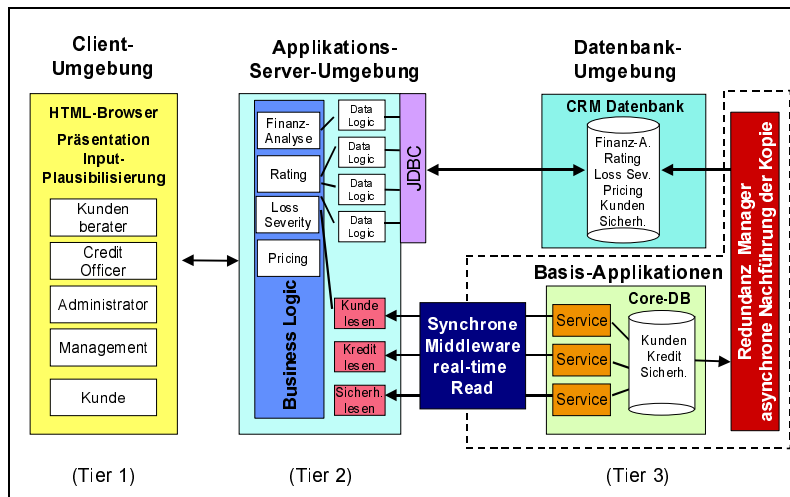
1. Dem Benutzer wird die Applikation ausschliesslich über einen Internet-Browser mit weitgehender HTML-Pur Oberfläche zur Verfügung gestellt. Damit wird die Komplexität vermieden, welche heute mit der Verbreitung jeder zusätzlichen Applikationskomponente auf der Seite des PC's der Endbenutzer zunimmt. Gleichzeitig ist die Applikation unabhängig von Betriebssystem und Plattform des Clients.
2. Bei der server-basierten Architektur ist keine SW-Verteilung auf eine hohe Anzahl Clients notwendig. Dadurch besteht die Möglichkeit, eine Applikation inkrementell und mit kurzen Auslieferungszeiten weiterzuentwickeln.
3. Durch die server-basierte Lösung ist die Stabilität insgesamt höher, da die Applikation auf dedizierten Server-Systemen installiert wird und so das Risiko möglicher Laufzeitkonflikte mit anderen Applikationen reduziert wird.

### **Design-Ziel 3: Integrationsfähigkeit**

Ein Kreditrisikomanagement-System ist in jeder Bank in ein bestehendes Applikations-Umfeld zu integrieren, um Partner-, Kredit- und Sicherheiten-Daten aus bereits vorhandenen Basis-Applikationen zu verwenden. Deshalb kommt der Integrationsfähigkeit in der System-Architektur grosse Bedeutung zu.

Aus verschiedenen Studien über die Integration von zugekaufter Standard-Software ist heute bekannt, dass der Aufwand für die Entwicklung von Schnittstellen und der Datenaustausch bis zu 35% des Gesamt-Aufwandes der Einführung einer neuen Applikation ausmacht. Deshalb gilt es, bereits im Design die Integration als Bestandteil der Lösung frühzeitig zu berücksichtigen, um auf diese Weise den Integrationsaufwand zu minimieren.

Grafik 3 zeigt die 3-Tier Architektur mit der Integration bestehender Basisapplikationen.



Grafik 3: Systemarchitektur und Integration

Generell bestehen zwei unterschiedliche technische Integrations-Anforderungen, welche in der System-Architektur abzudecken sind. Zum einen besteht die Anforderung, in real-time von den Kreditrisiko-Modulen auf die bestehenden Kundendaten, Kredite und Sicherheiten zuzugreifen (in Grafik 3 mit (1.) bezeichnet).

Auf der Seite der Kreditrisikomanagement-Module in Tier 2 können für diese Anforderung entsprechende isolierte Schnittstellen-Funktionen als Adapter vorgesehen werden. Diese greifen über verschiedene Middleware-Protokolle, wie z.B. CORBA oder andere Datenbank-Middleware Produkte synchron auf Services oder Daten aus den Basis-Applikationen zu. Muss das System mit unterschiedlichen Basis-Applikationen integriert werden, liegt somit die Integrationsarbeit nur in der Anpassung dieser Schnittstellen-Funktionen.

Aufgrund der modularen Architektur muss ein Ausschnitt bestehender Daten aus der Basisdatenbank (Core-Datenbank) redundant auf der Kreditrisiko-Datenbank (CRM-Datenbank) abgelegt werden. Dies führt zu einer maximalen Unabhängigkeit von den operativen Basisapplikationen und bietet die Möglichkeit, autonome Auswertungen auf der Basis der CRM-Datenbank durchführen zu können. Zudem sollten die Module in der Kundenberatung so eingesetzt werden, dass Daten für Interessenten und Neukunden zuerst in der CRM-Datenbank angelegt und erst bei Geschäftsabschluss in die Basisdatenbank übertragen werden.

Mit der gezielten Einführung von Daten-Redundanz verbunden ist die zweite Integrationsanforderung. Diese besagt, dass zur Sicherstellung der Datenkonsistenz alle Datenänderungen in den Basis-Applikationen asynchron und near real-time über einen Redundanz-Manager in der CRM-Datenbank nachzuführen sind (in Grafik 3 mit (2.) bezeichnet). Eine Transformation-Engine bietet sich als geeignete Lösung für das benötigte Mapping der Datenstrukturen an.

Als weitere Voraussetzung für das Erreichen der Integrationsfähigkeit ist ein flexibles Rollen- und Berechtigungs-Modell für die unterschiedlichen Benutzerrollen notwendig, da die Aufbau- und Ablauf-Organisationen von Bank zu Bank verschieden sein können. Mögliche Rollen im Kreditprozess sind, wie in der Clientumgebung in Grafik

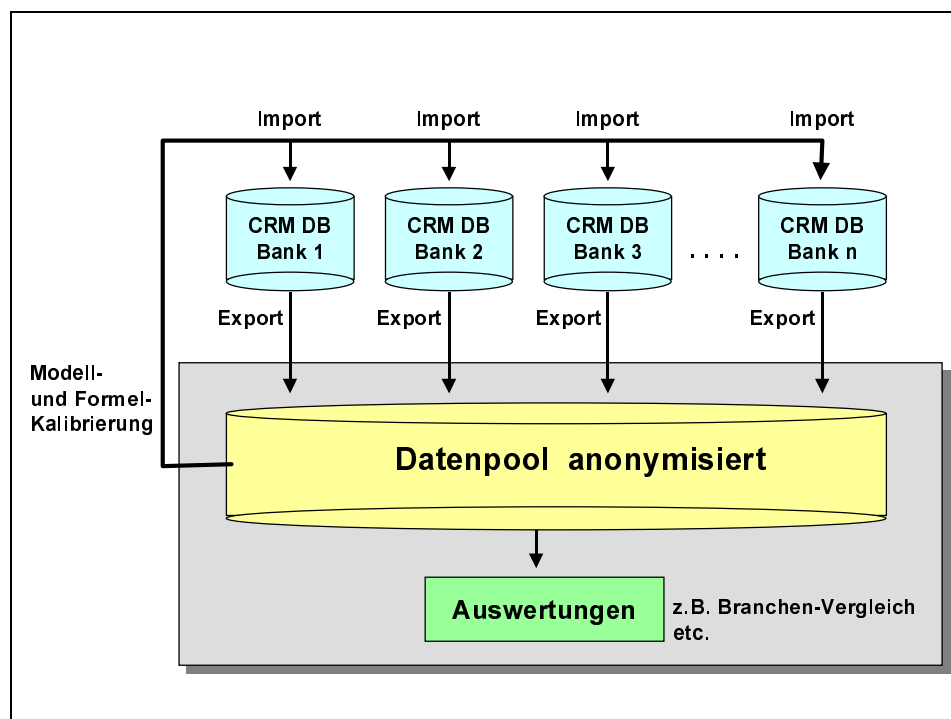
3 dargestellt, der Kundenberater, der Credit Officer, das Management, diverse administrative Funktionen bis hin zum Kunden selber im Falle eines Internet-Einsatzes.

#### Design-Ziel 4: Aufbau eines Datenpools

Eine kleine oder mittelgrosse Bank verfügt alleine nicht über die Menge an Kreditdaten, welche statistisch für die Kalibrierung eines Kreditrisikomodells notwendig wäre. Anzustreben ist deshalb, dass verschiedene Banken ihre Kreditdaten in einen gemeinsamen anonymisierten Datenpool einbringen, wie Grafik 4 zeigt. Wegen der grösseren Datenmenge führt ein solches Pooling von Daten zu einer verbesserten Schätzung der Kreditrisikomodelle, woraus jeder beteiligten Bank wieder ein Vorteil für ihr Kreditrisikomanagement erwächst.

Ein Datenpool ermöglicht zudem generelle Auswertungen über einen vergleichsweise grossen Datenbestand. Beispielsweise können Auswertungen bezüglich Branchen oder geographischer Lage von Kreditnehmern durchgeführt werden.

Als Designziel für die Systemarchitektur folgt, dass Import-Schnittstellen für die Rückführung der aus den statistischen Schätzungen im Datenpool gewonnenen Parameterwerten bereitgestellt werden. Ebenso sind Funktionen notwendig, die den Export und die Anonymisierung von Kreditdaten in den Datenpool ermöglichen. Die Aufdatierung des Datenpools findet typischerweise einmal im Quartal statt.



Grafik 4: Zusammenarbeit verschiedener Banken im Datenpooling

**Fazit:**

Die Natur von Kreditrisiken und die wegen der prekären Datenlage erschwerte Implementierung von Kreditrisikomodellen legen eine Zusammenarbeit von kleineren und mittelgrossen Banken im Bereich des Kreditrisikomanagements nahe. Der Systemarchitektur kommt dabei eine zentrale Bedeutung zu, denn nur eine offene Architektur mit modularem Aufbau erlaubt die effiziente Integration von Partnerbanken mit unterschiedlichen Basissystemen und Businessprozessen. Die heutige Webtechnologie bietet zudem über einen Thin-Client Ansatz einen Weg, die Einbindung verschiedenster Benutzergruppen aus ganz unterschiedlichen Betriebssystem- und Plattformumgebungen über Intranet, Extranet und Internet wesentlich zu erleichtern. Ein solches Vorgehen führt zur notwendigen technischen Flexibilität, welche eine Zusammenarbeit verschiedener Banken im Kreditrisikomanagement erst ermöglicht.