

Card Processing X.0

Serviceorientierte Architekturen als Weg zur Industrialisierung

In den letzten 20 Jahren hat sich die Verarbeitung des kartengestützten Zahlungsverkehrs erheblich gewandelt. Abwickler nutzen leistungsfähige Standardsysteme und kombinieren sie zu technisch anspruchsvollen, effizienten Systemlandschaften. Dieser Beitrag stellt aus der Historie heraus eine zukunftsfähige und einsetzbare Architektur vor, mit der Kartenzahlungen in Form von Services industrialisiert verarbeitet werden können.

Die richtige Systemlösung für das Processing im kartengestützten Zahlungsverkehr gehört für die Kartenindustrie zu den Schlüsselfaktoren, die über Wachstum, Innovationsfähigkeit und Wirtschaftlichkeit eines Leistungsanbieters entscheiden.

Der Kernsystem Ansatz

Bis in die 80er Jahre betrieben Herausgeber von Bankkarten und Abrechnungsdienstleister selbstentwickelte Anwendungen auf den im Haus vorhandenen Mainframe-Rechnern. Softwareanbieter entwickelten jedoch bereits Standard-Software für die Verarbeitung von Kartentransaktionen, die auf den aktuellen Transaktionsmonitoren und Dateisystemen basierten und sich damit modern und zukunftsfähig darstellten. In den 90er Jahren dominierten diese meist aus den USA kommenden Standard-Anwendungen den Markt für Kartenmanagementsysteme in Europa. Der Versuch, mit objektorientierten Ansätzen auf Windows/Intel Plattformen ein Kartensystem mit Geschäftsvorfällen, Ereignissen und Regeln zu erstellen, war letztlich nicht erfolgreich. Auch groß dimensionierte Processing Systeme mit eigener Laufzeit- und Entwicklungsumgebung konnten sich, selbst wenn sie auf relationaler Datenbanktechnologie aufsetzten, nur vereinzelt im Markt behaupten.

Vorherrschend blieben die etablierten, mit proprietären Programmiersprachen erstellten Systeme. Die IT-Architekten modularisierten ihre Software im Laufe des Lebenszyklus und trugen Einwänden gegen den monolithischen Ansatz damit formal Rechnung. Einzelne funktionale Systemkomponenten erhielten Namen, die jeweils auf Geschäftsbereiche des Processing verwiesen, doch die technische Architektur änderte sich ebenso wenig wie die Systemumgebung.

Der Satellitenansatz

Neben den nun in funktionale Module gegliederten Kernsystemen positionierten sich in der Systemlandschaft ab Mitte der 90er Jahre leistungsfähige Anwendungen vorwie-

gend für Spezialbereiche des Processing. Beispiele für diesen Trend sind Charge-back gekoppelt mit Workflow Management, Fraud Detection Systeme oder Archivierungssysteme. „Umsysteme“ satellitenartig an ein Kernsystem anzubinden, war in dieser Phase ein aktuelles Architekturprinzip. Die komfortablen Systeme erhielten damit fachliche Konkurrenz, doch in der Stammdatenhaltung, der Transaktionsverarbeitung und der Karten- und/oder Händlerabrechnung dominierten weiterhin die etablierten Lösungen.

Betriebswirtschaftliche Funktionen, wie die für Kartenzahlungsverarbeiter wichtige Abrechnung der Dienstleistung an die Banken, blieben rudimentär ausgestattet. Gleiches galt für die Überleitung von Zahlungsströmen in die eigene Buchhaltung, das Einnahmen- und Ausgaben-Monitoring und die Darstellung der Aufwands- und Ertragssituation bis in die GuV: Die Kernsysteme stellten zwar recht früh General Ledger Files als Schnittstelle bereit, diese erfüllten aber selten die Anforderungen der Fakturierung oder des Rechnungswesens.

Der Kompositionsansatz

Inzwischen setzen Kartenzahlungsverarbeiter in Kenntnis der Stärken und Schwächen ihrer technischen und kaufmännischen Anwendungen verstärkt auf einen heterogenen Ansatz: Sie wählen für jeden Bereich die beste Lösung¹ aus und integrieren diese zu einer Gesamtplattform. In Anbetracht überwiegender Vorteile werden die Nachteile dieser Verfahrensweise in Kauf genommen. Neben der Komponentenwahl für das technische und kaufmännische Geschäft gilt es, die Systemintegration zu bewältigen.

Vor allem Netzbetreiber und Kreditkarten-Acquirer nutzen den Kompositionsansatz und setzen ihre kaufmännischen Anwendungen als tragende Komponenten im Processing ein. Die zunehmende Leistungsfähigkeit kaufmännischer Standardsoftware begünstigt dieses Vorgehen. Customizing und Business Adds erlauben zudem eine gute Adaption an das Geschäft. Die Datenüberleitung in das eigene Rechnungswesen ist damit gelöst.

Da für Kundenportale, Prozesskontrolle, Data Warehousing und CRM ebenfalls Komponenten verfügbar sind, dürfte das Verfahren auch weiterhin genutzt werden. Weniger attraktiv ist der auf Standardanwendungen basierende Kompositionsansatz für Bankkartenherausgeber, die eine hohe Zahl von Konten verarbeiten müssen. Zur Integration können Bankkartenkonten beispielsweise innerhalb der Bankanwendung abgebildet werden. Unabhängig davon, ob ein Kernbankensystem Unterkonten in Form von Kreditkartenkonten funktio-

SERVICE ARCHITEKTUR



Ansatz für eine serviceorientierte IT-Architektur

Bildquelle: SYNGENIO

nal verwalten kann, ist aber genau zu prüfen, ob sich auch die progressiven Kartenprodukte innerhalb des Systems abbilden lassen.

Der X.0 Ansatz

In Zukunft dürften IT-Sichten, die auf Hard- und Software fokussiert sind, zunehmend obsolet werden: Das Handlungsfeld der IT bestimmen die unternehmerischen Geschäftsprozesse, die durch adäquate Services agil und skalierbar unterstützt werden müssen.

Services können intern oder von Dritten bezogen, durch Kauf oder in Lizenz erbracht werden. Sie lassen sich gegeneinander abgrenzen, die Vendoren sind austauschbar. Die neuen Bedingungen verlangen mehr organisatorisch und methodisch ausgebildetes IT-Personal. Gefordert sind Skills in den Bereichen IT-Servicemanagement, Providermanagement und Projektmanagement, sowie Qualifikationen in der Integration und im Schnittstellendesign. Serviceorientierte IT-Architekturmodelle sind Ausdruck einer von Konkurrenzdruck getriebenen Evolution. Ihre Bedeutung wächst, weil die Verarbeitung kartengestützter Transaktionen als Geschäftsfeld bei rückläufigen Erträgen steigenden Anforderungen unterliegt. Wie ein serviceorientierter Ansatz aussehen kann, zeigt die Grafik.

Services werden an unterschiedlichen Orten („Datacenter:“), von unterschiedlichen Softwareapplikationen („System:“)

erbracht und von unterschiedlichen Verantwortlichen gesteuert („Managed:“). Die Steuerung kann unternehmensintern, oder von einem Zulieferer erbracht werden. Der Gesamtverantwortliche muss die Entscheidung, ob ein Zulieferer einen Service in höherer Qualität oder günstiger erbringt als interne Unternehmenseinheiten, inhaltlich absichern. In dem gezeigten Blue Print einer serviceorientierten Architektur werden 15 Services mittels 11 Softwaresystemen an 8 unterschiedlichen Lokationen erbracht.

Unternehmen der Kartenzahlungsverarbeitung stellen ihren Geschäftsbetrieb vermehrt durch die Integration von Services dar. Die Folge ist eine Abkehr von spezifischen Entwicklungen zugunsten von Standardprodukten. Es zeigt sich auch, dass für Kontoführung, Abrechnung und andere zentrale Prozesse, die lange eine Domäne der Kernsysteme waren, zunehmend betriebswirtschaftliche Software eingesetzt wird.

Zusammenfassend verbindet sich mit dem X.0 Ansatz eine fortschreitende Standardisierung: Immer häufiger werden Kernprozesse mit erprobten Softwareprodukten von Rechenzentrums-Dienstleistern erbracht, die von unterschiedlichen Standorten aus Kunden diverser Branchen versorgen. Es entstehen industrielle IT-Fertigungen mit segmentierten Prozessen, die intern oder extern verantwortet und auf mehrere Produktionsorte verteilt sein können. Das System wird über eine überwachte Logistikkette integriert, die Zulieferer sind austauschbar. Payment Service Provider decken bereits ein breites Fertigungsspektrum ab und integrieren unterschiedliche Bezahlprodukte horizontal. Die Granularität der Segmentierung von Services wird im Verlauf der X.0 Entwicklung eher feiner werden. Absehbar ist auch, dass die großen Abwickler neue „Next Generation Finance“-Produkte in Form entsprechender Services integrieren und sich im Interesse von Agilität und Skalierbarkeit industrieller aufstellen werden. ■

1) Anglo-amerikanisch auch „Best of Breed“ genannt.



Autor:
Peter Hessler
Senior Business Consultant,
SYNGENIO AG