

12 Beispiel einer Architekturdokumentation

Dieses Kapitel dokumentiert die Architektur eines realen Softwaresystems gemäß den methodischen und strukturellen Ansätzen aus Kapitel 4. Es orientiert sich dabei an der arc42-Strukturvorlage, die ich Ihnen in Abschnitt 4.9.1 vorgestellt habe.

Anmerkung zur Nummerierung: Hier habe ich die „normale“ Überschriftenzählung dieses Buches außer Kraft gesetzt und nummeriere die Abschnitte exakt so, wie es in einer *echten* Architekturdokumentation der Fall wäre.

1 Einführung

Zweck des Systems

Dieses Dokument beschreibt die Software-Architektur des M&M¹-Systems zur Migration von ca. 20 Millionen Personen- und Kontodaten einer Organisation aus der Finanzdienstleistung.

Leserkreis

- Alle in Abschnitt 2.3 dieser Dokumentation genannten Stakeholder des M&M-Systems.
- Software-Entwickler, Architekten und technische Projektleiter, die Beispiele für eine Architekturdokumentation (auf Basis des arc42-Templates) suchen.

¹ M & M steht für **M**igration von **M**assendaten.

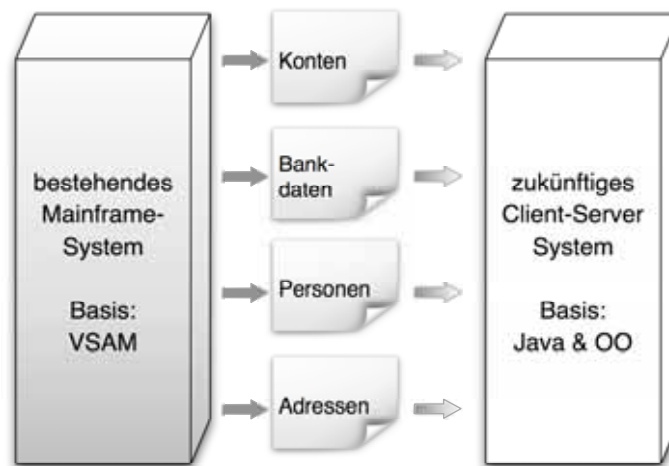


Abbildung 12.1 Zweck des Systems: Migration bestehender Daten

- Mitarbeiter von IT-Projekten, die sich das Leben in der Software-Architektur vereinfachen, indem sie auf ein bewährtes Template zur Dokumentation zurückgreifen.
- Solche, die immer noch glauben, Templates wären Teufelswerk und Architekturen gehörten auf die Rückseite gebrauchter Briefumschläge. Die können hier lesen, wie es anders geht.

2 Architekturtreiber

Die wichtigsten funktionalen Anforderungen lauten wie folgt:

- Bestehende Kunden-, Konto- und Adressdaten sollen von den bisherigen VSAM²-basierten Mainframe³-Programmen aus dem EBCDIC-Format in das Objektmodell einer in Entwicklung befindlichen Java-Anwendung (Unicode/ASCII) migriert werden.
- Die Ausgangsdaten liegen in Form verschiedener Dateien oder „Bänder“ vor, deren Einträge nach fachlichen Kriterien einander zugeordnet werden müssen.
- Früher waren die Daten nach Konten organisiert, im neuen System sind Personen der Bezugspunkt.

² VSAM = Virtual Storage Access Method; eine Zugriffsmethode für Dateien, die von IBM-Großrechnern verwendet werden. VSAM-Dateien bestehen aus einem Metadatenkatalog sowie mindestens einer physischen Datei. Mehr zu VSAM in den Literaturhinweisen am Ende dieses Kapitels. Übrigens gibt es das Gerücht „VSAM ist grausam“.

³ Eine aktuelle Einführung in OS/390 finden Sie unter <http://www.informatik.uni-leipzig.de/cs/esvorles/index.html>.

- Teile der früheren Schlüssel- oder Kennzahlensysteme sollen in neue Schlüssel überführt werden.

2.1 Architekturziele

Die primären Architekturziele von M&M lauten:

- Effizienz (Performance): Migration von ca. 20 Millionen Personen- und Kontodaten innerhalb von maximal 24 Stunden.
- Korrektheit: Die Migration muss revisionssicher und juristisch einwandfrei sein. Hierzu sind umfangreiche Maßnahmen zur Fehlervermeidung und -erkennung nötig.

Nicht-Ziele

Was M&M nicht leisten soll:

- Änderbarkeit oder Flexibilität der fachlichen Transformationsregeln – die Migration ist einmalig.
- Es bestehen keine besonderen Anforderungen an Sicherheit – M&M wird nur ein einziges Mal produktiv betrieben, und das innerhalb eines gesicherten Rechenzentrums.

2.2 Geschäftskontext

Der Auftraggeber von M&M, die Firma „Fies Teuer AG“, betreibt seit Jahren einige Host-Anwendungen zur Pflege von Personen- und Kontodaten.

Fies und Teuer AG hat in Deutschland ca. 20 Millionen Kunden und pflegt für diese Kunden insgesamt mehr als 50 Millionen Konten.

Kunden können natürliche oder juristische Personen sein, teilweise auch andere Organisationen (Verbände, Vereine etc.).

Konten enthalten neben reinen Buchungsinformationen auch statistische oder sonstige finanzbezogene Informationen zu den Kunden oder mit ihnen assoziierten Personen oder Organisationen.

Die inhaltliche Bedeutung dieser Konto- und Buchungsinformationen sowie die darin enthaltenen Attribute haben über die Betriebszeit der bisherigen Anwendung stark gewechselt. Zur Umstellung auf ein neues und konsolidiertes Objektmodell hat ein Team von ca. 20 Fachexperten eine Menge von deutlich über 100 fachlichen Regeln aufgestellt, nach denen die Migration ausgeführt werden muss.

2.3 Stakeholder

- Management der Firma „Fies und Teuer AG“, das eine reibungslose und fehlerfreie Migration wünscht.
- 20 Millionen Kunden der Firma „Fies und Teuer AG“, denen die korrekte Migration ihrer finanzbezogenen Informationen wichtig ist. Diese Stakeholder treten im Projektverlauf überhaupt nicht in Erscheinung, sondern sind nur indirekt beteiligt.
- Formale Revision oder Buchprüfung, die auf die juristische und buchhalterische Korrektheit der Migration achtet.
- Die Boulevardpresse, die auf „Fies und Teuer AG“ ein besonderes Augenmerk legt. Jegliches Fehlverhalten des Unternehmens wird gnadenlos publiziert :-)

3 Einflussfaktoren

3.1 Technische Einflussfaktoren

Tabelle 12.1 Technische Einflussfaktoren

Randbedingung	Erläuterung
Hardware-Infrastruktur	IBM Mainframe als Plattform des Altsystems, Sun Solaris Cluster für das Zielsystem
Software-Infrastruktur	<ul style="list-style-type: none"> ▪ Sun Solaris als Betriebssystem der Zielumgebung ▪ Oracle als neue Datenbank ▪ J2EE kompatibler Applikationsserver als mögliche Betriebsumgebung
Systembetrieb	Batch
Grafische Oberfläche	keine, Bedienung kann per Konsole erfolgen
Bibliotheken, Frameworks und Komponenten	Beliebig
Programmiersprachen	Aufgrund des vorhandenen Know-hows des Entwicklungsteams soll in Java programmiert werden.
Referenzarchitekturen	keine vorhanden
Analyse- und Entwurfsmethoden	objektorientiertes Vorgehen
Datenstrukturen	Objektmodell der Zielumgebung bekannt (Bestandteil eines anderen Projektes)
Programmierschnittstellen	Daten werden durch (bestehendes) Host-System geliefert.
Programmiervorgaben	Keine
Technische Kommunikation	mit Host per ftp oder sftp

3.2 Organisatorische Einflussfaktoren

„Fies und Teuer AG“ ist als juristisch penibler Auftraggeber bekannt. Das Management versteht rein gar nichts von IT (und gerüchteweise auch nichts von Geld, aber das ist eine andere Geschichte).

Die verworrenen Eigentumsverhältnisse von Fies und Teuer AG lassen komplizierte Entscheidungsstrukturen befürchten.

Fies und Teuer AG hat langfristige Lieferverträge mit diversen IT-Dienstleistern abgeschlossen, die eine marktorientierte und freie Auswahl eventuell benötigter externer Mitarbeiter methodisch verhindern.

Eine hochgradig effektive externe Qualitätssicherung fordert eine umfangreiche Dokumentation (und prüft diese sogar auf inhaltliche Korrektheit – in der IT-Branche eher unüblich, aber das ist ebenfalls eine andere Geschichte).

4 Kontextsicht

Inhalt

Die Kontextsicht grenzt das System von allen Nachbarsystemen ab und legt damit die wesentlichen externen Schnittstellen fest.

Fachlicher Kontext

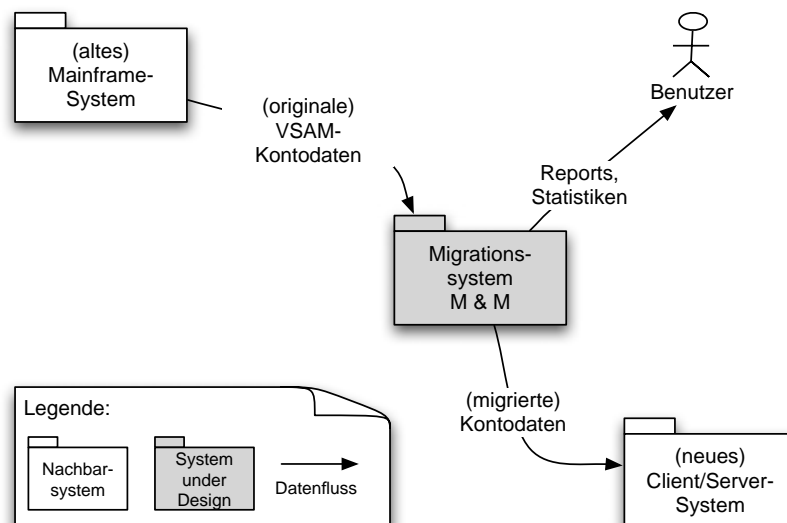


Abbildung 12.2 Fachlicher Kontext

Verteilungskontext

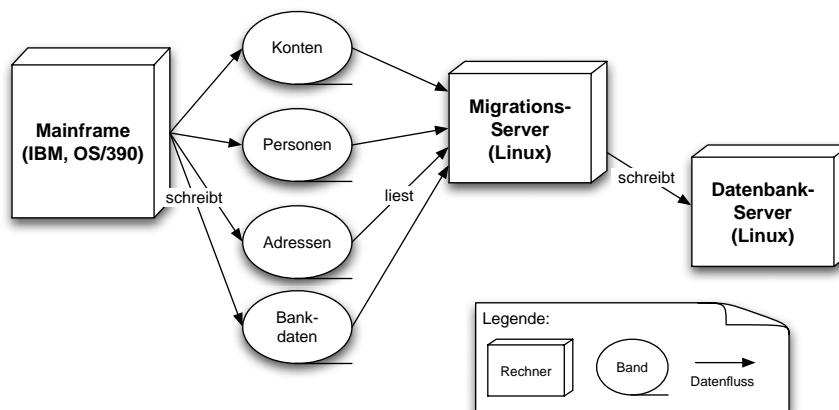


Abbildung 12.3 Verteilungskontext

Die Migration wird auf zwei getrennten Servern ablaufen: ein Migrationsserver zur eigentlichen Datenmigration und Ausführung der technischen und fachlichen Transformationen, ein zweiter als Datenbankserver.

Das neue System (in der Abbildung nicht dargestellt) wird direkt auf die von der Migration erstellte Datenbank zugreifen. Der Migrationsserver muss daher das künftige Klassen- und Tabellenmodell kennen.

5 Bausteinsicht

Ausgehend von einer Menge von VSAM-Dateien (geliefert als Bänder) konvertiert („migriert“) die M&M-Anwendung sämtliche Datensätze in das neue Objektmodell.

5.1 M&M Bausteinsicht Level 1

Grundlegende Strukturentscheidung: Pipes-und-Filters

Zur Lösung des M&M-Migrationsproblems wenden wir ein modifiziertes Pipes-und-Filter-Architekturmuster an, ausführlich beschrieben im Buch „Pattern-Oriented Software Architecture“ von Buschmann et.al.

Ein wesentliches Entscheidungskriterium für diese Architektur war die Möglichkeit, Teile dieser Bausteine im Bedarfsfall (Performancebedarf) parallelisieren zu können.

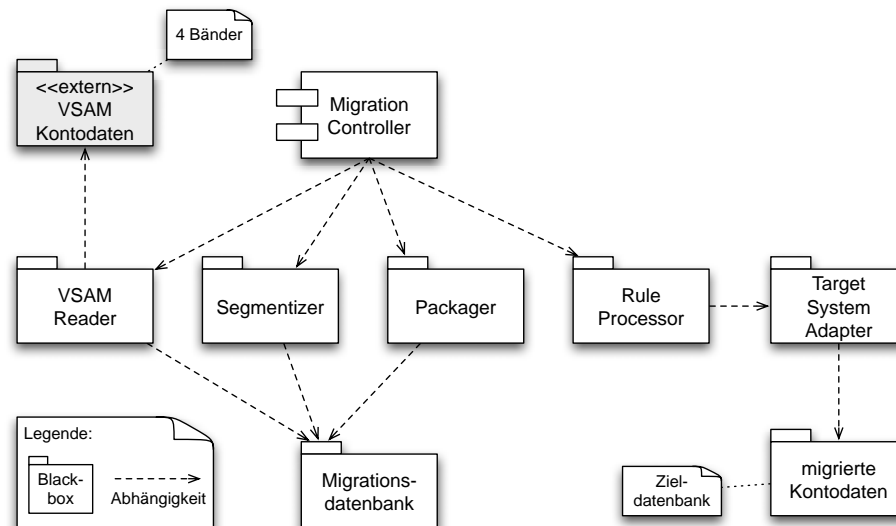


Abbildung 12.4 Whitebox-Darstellung des Migrationssystems, Level-1

Die folgenden Abschnitte beschreiben die in der Abbildung dargestellten Bausteine:

- Migration Controller
- VSAM Reader
- Segmentizer
- Migrationsdatenbank
- Packager
- Rule Processor
- Target System Adapter

5.1.1 Migration Controller

Der Migration Controller koordiniert den Ablauf der Migration. Es handelt sich um eine einzelne Java-Klasse mit ausführlicher Ausnahme- und Fehlerbehandlung sowie einer Schnittstelle für Reporting und Auswertungen (im Diagramm nicht dargestellt).

5.1.2 VSAM Reader

- *Zweck/Verantwortlichkeit:* Konvertiert Datensätze aus VSAM-Format (EBCDIC-Codierung) in ein unter Unix direkt verarbeitbares ASCII- oder UNICODE-Format. Gleichzeitig werden die Bestandteile der VSAM-Daten (teilweise besitzen einzelne Bits- oder Bitgruppen in der EBCDIC-Darstellung besondere Bedeutung) in einzeln identifizierbare Datenelemente aufgelöst.

Der VSAM Reader führt keinerlei fachliche Prüfungen durch, sondern lediglich Formatkonvertierungen.

- *Schnittstelle(n)*: Eingabe von VSAM-Dateien. Für jede Dateiart (Personen, Kontendaten, Adressdaten, Bankdaten) existieren jeweils einzelne Reader-Komponenten, die den Satzaufbau ihrer jeweiligen Eingabedateien kennen.
- *Variabilität*: keine, weil das Gesamtsystem nur einmalig ablaufen soll
- *Leistungsmerkmale*: <entfällt>
- *Ablageort/Datei*: <entfällt>
- *Sonstige Verwaltungsinformation*: keine
- *Offene Punkte*: Für eine spätere Nutzung bei anderen Unternehmen könnten die einzelnen Konverter „pluggable“ gemacht werden. Dies ist zurzeit nicht vorgesehen.

5.1.3 Segmentizer

- *Zweck/Verantwortlichkeit*: Vorbereitung der Parallelisierung des „Rule Processors“: Zuordnung der einzelnen Datensätze der unterschiedlichen Datenquellen (Personendaten, Kontendaten, Bankdaten, Adressdaten) zu zusammengehörigen Datengruppen oder Segmenten. Elemente eines Segments können völlig unabhängig von Inhalten anderer Segmente bearbeitet werden. Notwendig ist die Aufteilung der Eingangsdaten auf mindestens 3-5 verschiedene Segmente.
- *Schnittstelle(n)*:
 - *Eingang*: Die vom VSAM Reader konvertierten Datenfelder der Eingangsdaten.
 - *Ausgang*: Migrationsdatenbank
- *Variabilität*: keine
- *Leistungsmerkmale*: Es geht um eine möglichst *schnelle* Einteilung in Segmente, nicht um Optimierung oder Balancierung der Segmente. Daher liegt das Hauptaugenmerk auf Performance.
- *Ablageort/Datei*: entfällt im Beispiel.
- *Sonstige Verwaltungsinformation*: keine
- *Offene Punkte*: Segmentierungskriterien zurzeit nicht konfigurierbar, sondern ausprogrammiert.

5.1.4 Migrationsdatenbank

Speichert die konvertierten und segmentierten Migrationsdaten zwischen. Dient im Sinne des Pipe&Filter-Architekturmusters als Pipe, d.h. enthält keinerlei eigene Logik. Die Migrationsdatenbank ist eine Menge von Tabellen einer relationalen Datenbank (Person, Konto, Bank, Adresse plus einige Schlüsseltabellen).

5.1.5 Packager

Siehe Rule Processor im folgenden Abschnitt.

5.1.6 Rule Processor (und Packager)

Zweck/Verantwortlichkeit: Erzeugt einen Objektgraphen mit der jeweiligen Person (natürliche oder nichtnatürliche Person) als Wurzelknoten sowie allen zugehörigen fachlichen Objekten als Zweigen (Adressen, Bankverbindungen, Konten, Buchungen).

Der Rule Processor führt damit den eigentlichen fachlichen Migrationsschritt aus der alten (datenartorientierten) Darstellung in die neue (personenorientierte) Darstellung durch. Eventuell sind hierzu mehrere Iterationen über die Regelbasis notwendig.

Beispiele (sind Bestandteil der Anforderungsspezifikation; hier zur Illustration des notwendigen Vorgehens erneut dargestellt):

- Bei der Migration einer verheirateten Frau müssen vorher der Ehepartner sowie gegebenenfalls sämtliche früheren (geschiedenen, verschwundenen oder gestorbenen) Ehepartner migriert werden.
- Bei der Migration von Minderjährigen müssen vorher die natürlichen Eltern, Vormunde und Pflegeeltern migriert werden.
- Bei der Migration von Firmen müssen vorher sämtliche Vorgängerunternehmen, aus denen die aktuelle Firma hervorgegangen ist, migriert werden. Gleiches gilt analog für Vereine oder Verbände.

Damit diese fachlich zusammengehörigen Daten oder Datengruppen durch parallel ablaufende Prozesse migriert werden können, müssen in einem Vorverarbeitungsschritt sämtliche zusammengehörigen Daten als „Paket“ zusammengestellt werden. Diese Aufgabe übernimmt der Packager.

5.1.7 Target System-Adapter

Zweck/Verantwortlichkeit: übernimmt den vom Rule Processor erzeugten Objektgraphen in das Objektmodell des (neuen) Zielsystems. Hierfür sind eventuell kleinere Modifikationen notwendig, je nach endgültig entschiedener Laufzeitumgebung auch nur die abschließende Persistierung.

Nach erfolgreicher Übernahme in das neue System (Zielsystem) liefert der Target System-Adapter eine Meldung zurück – im Fehlerfall wird der gesamte Objektgraph zusammen mit den Fehlermeldungen des Zielsystems in die Fehlerdatenbank geschrieben.

5.1.8 Migrierte Kontodaten in Zieldatenbank

Eine relationale Datenbank in der Struktur des neuen Client-Server-Kontosystems. Die endgültige Tabellenstruktur wird erst im Verlauf der Entwicklung feststehen und ausschließlich durch das (parallel laufende) Projekt „Kontosystem-Neu“ definiert.

Sie wird durch den Target System-Adapter gefüllt.

5.2 M&M Bausteinsicht Level 2

Dieser Abschnitt beschreibt die Blackbox-Bausteine der Ebene 1 im Detail. Aufgrund des Beispielcharakters dieser Dokumentation sind nur wenige Aspekte detailliert dargestellt.

5.2.1 VSAM-Reader Whitebox

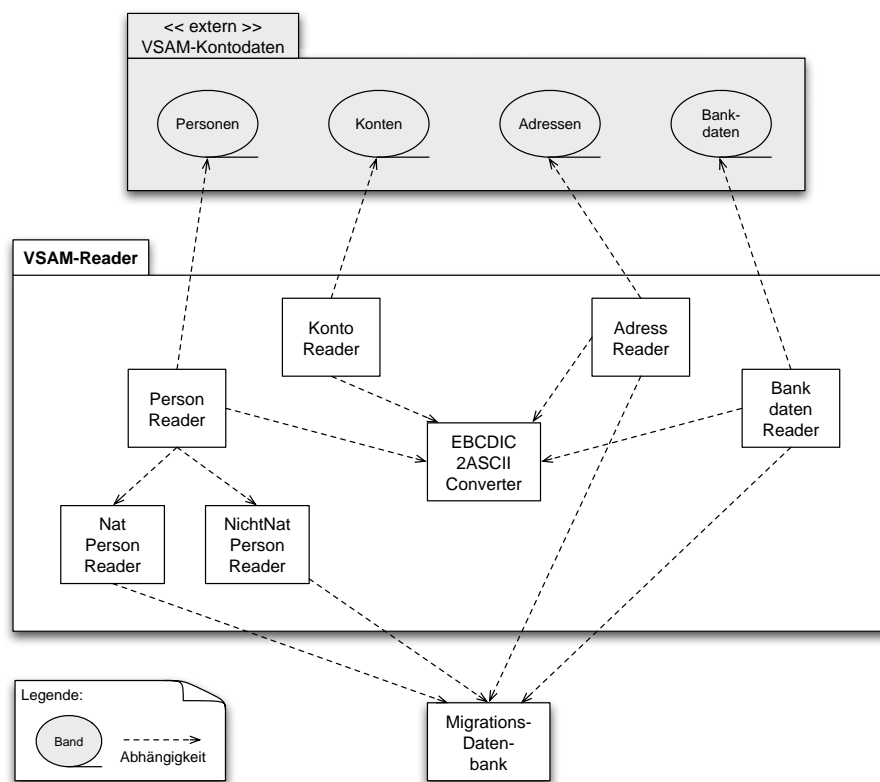


Abbildung 12.5 Interner Aufbau des VSAM-Readers

Konzeptionell sauberer wäre eine Trennung dieses Bausteins in den EBCDIC-ASCII-Konverter sowie die einzelnen spezifischen Reader-Bausteine gewesen.

Diese wurde vom Entwicklungsteam als Überstrukturierung angesehen und daher aus pragmatischen Gründen verworfen.

Die Beziehungen zwischen den Reader-Bausteinen und der EBCDIC2ASCII-Converter-Komponente sind reine Benutzt-Beziehungen. Die einzelnen Reader schreiben ihre Ergebnisse in die Migrationsdatenbank.

5.2.1.1 Person-, Konto-, Adress-, BankdatenReader

Die Reader-Komponenten lesen jeweils eine bestimmte Datenart aus „ihrem“ Eingabeband. Sie überführen sämtliche darin enthaltenen Daten in eigenständige Attribute.

Die einzelnen Reader kapseln die Syntax der „alten“ VSAM-Repräsentation sowie die technischen Details der Datencodierung.

5.2.1.2 EBCDIC2ASCII Converter

- Zweck/Verantwortlichkeit: Konvertiert eine Zeichenkette von EBCDIC nach ASCII.
- Schnittstelle(n):
- Eingabe: Zeichenkette, maximale Länge 256 Zeichen
- Ausgabe: Zeichenkette, maximale Länge 256 Zeichen
- Offene Punkte: Erweiterung auf längere Zeichenketten (bei M&M nicht erfordert, darum entfallen).

5.2.2 Rule Processor Whitebox

Der Rule Processor führt die eigentliche fachliche Migration von Datensätzen durch. Er kombiniert die unterschiedlichen Kategorien von Eingangsdaten (Personen, Konten und Buchungen, Adresse, Bankverbindungen) miteinander und bearbeitet sämtliche fachlichen Sonderfälle. Insbesondere besorgt der Rule Processor die benötigte Reorganisation der Daten, so dass zukünftig die Personen als Einstiegspunkte in die Navigation dienen, statt wie früher die Konten.

Strukturelevante Entwurfsentscheidungen:

- Kein kommerzieller Regelinterpreter: Nach einigen Versuchen mit kommerziellen Regelmaschinen entschied sich das Projektteam gegen den Einsatz einer solchen Komponente. Letztendlich war der Grund, dass die Umsetzung der fachlichen Regeln in eine Regelsprache keine wesentliche Vereinfachung gegenüber der Programmierung in Java gebracht hätte.

Anmerkung: Aus heutiger Sicht (also posthum...) würde ich diese Entscheidung noch mal überdenken und insbesondere Open-Source-Regelframeworks⁴ auf ihre Tauglichkeit hin überprüfen.

⁴ Wie beispielsweise JBoss-Drools (<http://labs.jboss.com/drools>).

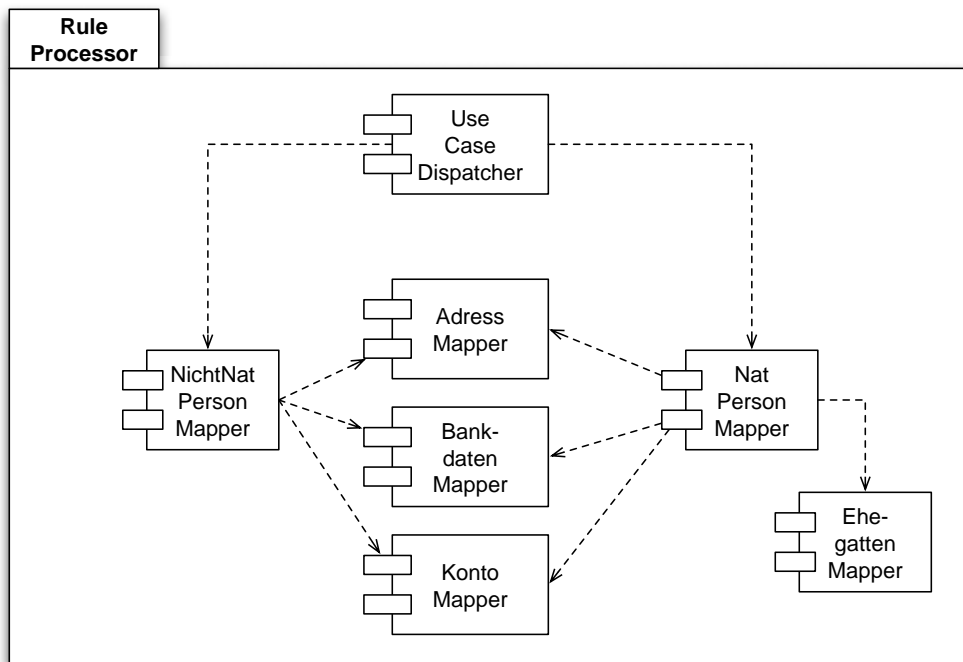


Abbildung 12.6 Interner Aufbau des Rule Processors

- Keine explizite Regelsprache: Der Ansatz, die Regeln von den Fachexperten formulieren und nur durch einen Interpreter ausführen zu lassen, scheiterte an der Komplexität des gesamten Datenumfelds. Die effiziente und performante Navigation innerhalb des jeweils benötigten Objektraumes war mit einer Regelsprache nicht zu bewerkstelligen. Siehe dazu die Anmerkung zum Regelerpreter.

Anmerkung: Aufgrund des Beispielcharakters dieser Dokumentation sind nachfolgend nur der UseCaseDispatcher sowie der PersonMapper ansatzweise beschrieben. In einer echten Dokumentation sollten Sie alle Bausteine soweit wie nötig detaillieren.

5.2.2.1 UseCase Dispatcher

Zweck/Verantwortlichkeit: Der UseCase Dispatcher entscheidet aufgrund der Datenkonstellation, welche der fachlichen Mapper-Komponenten angesprochen werden muss. Diese Entscheidung ist nicht trivial und nicht durch die Mapper selbst zu treffen. Teilweise müssen neben den aktuellen Daten weitere Sätze gelesen werden, um die Entscheidung abschließend treffen zu können.

Beispiele:

- Einpersonengesellschaften als nichtnatürliche Personen können teilweise durch den Personen-Mapper abgebildet werden, abhängig vom Gründungsdatum der Gesellschaft, dem Verlauf der Firmengeschichte (muss immer eine Einpersonengesellschaft gewesen sein und darf nicht aus einer Mehrpersonengesellschaft hervorgegangen sein).
- Natürliche Personen, die in der Vergangenheit Mitinhaber von Mehrpersonengesellschaften gewesen sind und darüber hinaus heute noch als Unternehmer tätig sind, müssen in Abhängigkeit von der Rechtsform der früheren Gesellschaft sowohl als natürliche als auch als nichtnatürliche Person geführt werden. Die Konto- und Buchungsinformationen sind entsprechend aufzuteilen.

5.2.2.2 PersonMapper

Zweck/Verantwortlichkeit: Erzeugt aus den Eingabedaten eine natürliche Person im neuen Objektmodell und füllt die Adressdaten, Bankdaten, Konto- und Buchungsdaten entsprechend auf. Handelt es sich um einen Verheirateten, so wird die zugehörige Ehefrau (sowie eventuell Ehefrauen aus früheren Ehen) passend migriert.

Der PersonMapper benutzt die anderen Mapper.

<<weitere Architekturbausteine werden in diesem Beispiel nicht dokumentiert>>

6 Laufzeitsicht

In der ersten Phase lesen die einzelnen Reader des VSAM-Readers alle angelieferten Dateien („Bänder“) und befüllen die Migrationsdatenbank. Danach teilt der Segmentizer die Daten in parallel bearbeitbare Segmente auf.

Der RuleProcessor lässt durch einen Packager (in der Abbildung – auf der nächsten Seite – durch den Aufruf „getPackage“ gekapselt) alle für die Migration einer Person oder eines Kontos notwendigen Daten aus der Migrationsdatenbank selektieren und führt anschließend die fachlichen Regeln auf diesem Datenpaket („Package“) aus.

Der TargetSystemAdapter schreibt die migrierten Daten in die neue Datenbank.

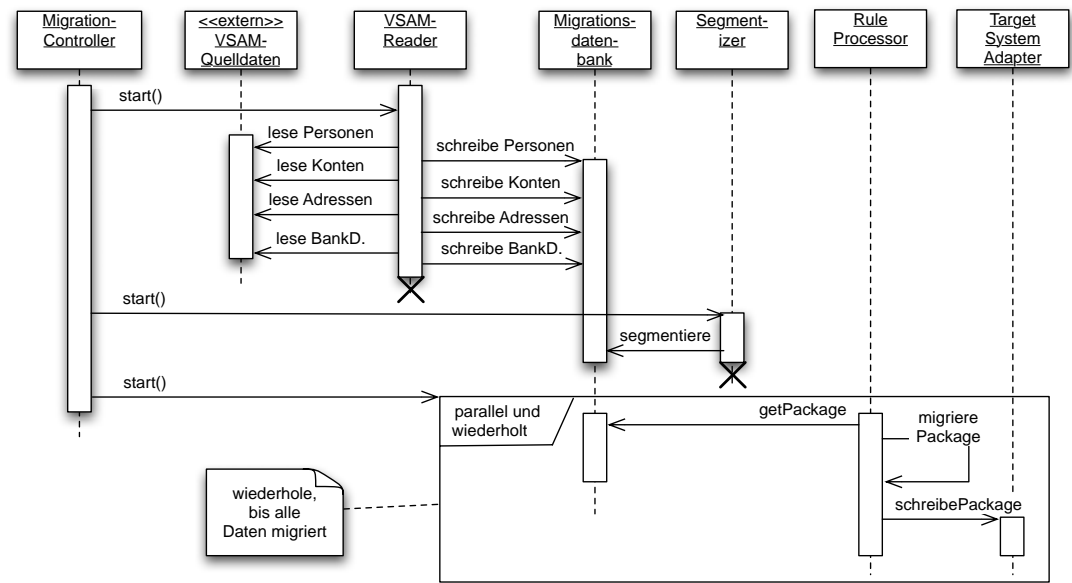


Abbildung 12.7 Laufzeitsicht der M&M-Migration

7 Verteilungssicht

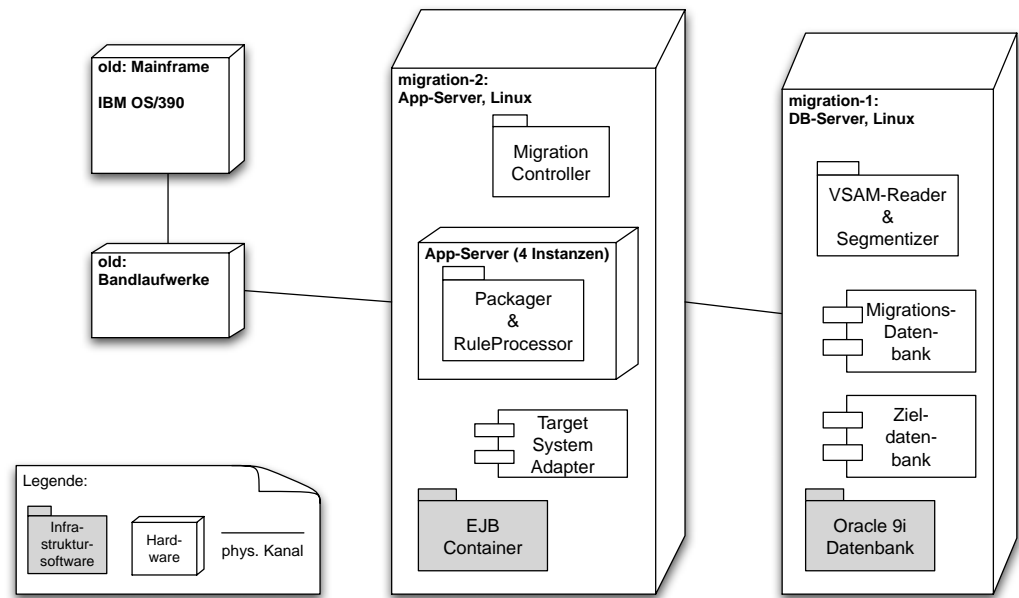


Abbildung 12.8 Verteilungssicht: Migration auf zwei getrennten Servern

Die gesamte Migration findet auf einem eigenständigen Linux-Server („migration-2“) statt, auf dem in einem EJB-Container mehrere parallele Instanzen der eigentlichen Migrationskomponenten (Packer & RuleProcessor) ablaufen.

Über einen schnellen Bus ist dieser Server mit dem DB-Server („migration-1“) verbunden, auf dem der VSAM-Reader & Segmentizer die initiale Befüllung der Migrationsdatenbank vornehmen.

Überraschend, jedoch vom Auftraggeber so entschieden: Die Bandlaufwerke mit den vom bestehenden Mainframe erzeugten VSAM-Dateien werden an den Server „migration-2“ angeschlossen.

Anmerkung: In einer echten Architekturdokumentation sollten Sie hier weitere Leistungsdaten der Hardware (Rechner, Netze, Bus-Systeme etc.) beschreiben.

8 Entwurfsentscheidungen

Inhalt: Die wesentlichen Entwurfsentscheidungen und ihre Gründe.

- Aufgrund der hohen Performanceanforderungen werden Datenbank und Migrationslogik auf zwei getrennten Servermaschinen ausgeführt. Als DB-Server kommt die später vom neuen Kontosystem genutzte Maschine zum Einsatz.
- Die Verarbeitung der fachlichen Regeln durch den RuleProcessor (siehe Abschnitt 5.1.6 beziehungsweise 5.2.2) wird durch mehrere parallele Prozesse (innerhalb des J2EE Application Servers) durchgeführt. Eine rein sequentielle Bearbeitung aller vorhandenen Personen wäre in der verfügbaren Laufzeit von 24 Stunden nicht möglich gewesen.

9 Szenarien zur Architekturbewertung

<Entfällt im Beispiel>

10 Architekturaspekte

10.1 Persistenz

Die Datenspeicherung im neuen System übernimmt ein Java Framework auf der Basis von J2EE, genauer gesagt Bean-Managed Persistence. Das Mapping der Objektstruktur auf die relationale Datenbank wird vom Framework komplett übernommen.

Für die eigentlichen Migrationskomponenten wird Persistenz damit nahezu transparent: Ist ein Objektgraph im Speicher aufgebaut, so kann er auf einmal gespeichert werden.

10.2 Ablaufsteuerung

Die Koordination und Ablaufsteuerung aller Migrationsprozesse wird durch einen zentralen Baustein (MigrationController) durchgeführt.

Die gesamte Verarbeitung erfolgt satzweise. Die Personen-Datei wird satzweise verarbeitet, sämtliche zugehörigen Kontoinformationen werden diesen Sätzen zugeordnet und die resultierenden Datenpakete anschließend gemeinsam migriert.

Aus Kostengründen sollte keine kommerzielle Workflow-Engine eingesetzt werden.

10.3 Ausnahme- und Fehlerbehandlung

Aufgrund der hohen Korrektheitsanforderungen der Auftraggeber werden sämtliche Ausnahmen beziehungsweise nicht migrierbaren Datensätze in einer Fehlertabelle gespeichert.

Sämtliche Fehlersätze müssen nach der Migration von Sachbearbeitern manuell migriert werden, und dafür stehen lediglich ca. 200 Personentage Aufwand zur Verfügung. Jeden Tag kann ein Sachbearbeiter im Mittel 25 Personen migrieren, daher dürfen in der Fehlertabelle höchstens 5000 ($= 20 * 25$) Datensätze enthalten sein.

10.4 Transaktionsbehandlung

Die einzelnen Migrationsschritte speichern ihren Zustand jeweils satzweise ab. Dadurch ist zu jedem Zeitpunkt ein Wiederaufsetzen durch Neustart der gesamten Anwendung möglich.

11 Projektaspekte

<entfällt im Beispiel>

12 Glossar und Referenzen

Das Glossar entfällt im Beispiel.

Mehr zu VSAM im frei verfügbaren IBM-Redbook:

<http://www.redbooks.ibm.com/abstracts/sg246105.html>