

Der Nutzen von Interimsarchitekturen bei strategischen Transformationsmaßnahmen des Architekturmanagements

Johannes Bach, Christoph Ringelstein, Klaus Zitzmann

Architekturmanagement
Debeka-Gruppe
Ferdinand-Sauerbruch-Str. 18
56058 Koblenz

{johannes.bach,christoph.ringelstein,klaus.zitzmann}@debeka.de

Abstract: Im Rahmen der Aufgaben des Architekturmanagements wurden Defizite in der Systemarchitektur der Debeka-Gruppe identifiziert. Aufgrund der Komplexität der Anwendungs- und Systemlandschaft sowie der bereits getätigten Investitionen werden unter Abschätzung von Risiken und Kosten sogenannte Interimsarchitekturen propagiert, die eine schrittweise Transformation unter Einbeziehung der Organisation während des laufenden Betriebes ermöglichen.

1 Ausgangssituation

Das Architekturmanagement macht das Zusammenwirken des Geschäfts mit der IT ganzheitlich, nachhaltig, systematisch und methodisch sichtbar und beherrschbar [Ha12]. Es wirkt, indem es die IT-Architektur laufend im Ist und Soll pflegt bzw. dokumentiert, sie analysiert und Defizite aufzeigt sowie Transformationsmaßnahmen plant, aber auch laufende Vorhaben und Projekte begleitet und berät [OG11].

Die IT-Landschaft der Debeka-Gruppe ist geprägt durch siloartige Anwendungen, die mit unterschiedlichsten Technologien implementiert wurden (siehe Abbildung 1: Ausgangsarchitektur). Die Anwendungen sind untereinander bidirektional integriert. Das Gesamtsystem ist dadurch heterogen und wird immer schwerer beherrschbar. Zusätzlich erschweren Altlasten einer vorhergehenden Plattformmigration die Situation.

Im Laufe der Analyse des Architekturmanagements wurde deutlich, dass diese Defizite nur durch einen Paradigmenwechsel beseitigt werden können. Aufgrund der großen Komplexität wurde aber schnell deutlich, dass dies nur schrittweise erfolgen kann, zumal der Betrieb des Tagesgeschäfts erhalten bleiben muss.

2 Idealbild und Interimsarchitektur

Der Paradigmenwechsel wird durch ein fernes Idealbild beschrieben, dessen Kernelemente den Mustern einer service- und prozessorientierten Architektur gehorchen. Dieses

Idealbild wird nie erreicht, dient aber dem IT-Management als Leitbild für seine zukünftigen Entscheidungen. Dem gegenüber steht die sogenannte Interimsarchitektur (siehe Abbildung 2: Interimsarchitektur), welche als erste realistisch zu erreichende Approximation an das Idealbild dient und unter Beachtung der gegebenen Altlasten als erster Schritt implementiert werden kann. Durch diese Beschränkung können der Umfang der Maßnahmen und deren Risiko begrenzt werden.

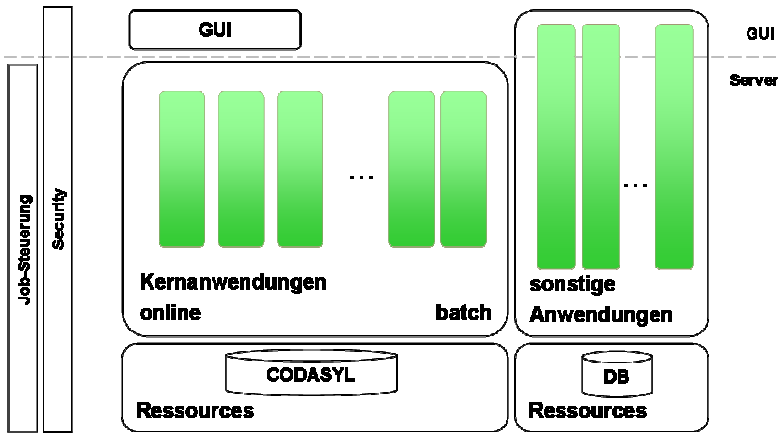


Abbildung 1: Ausgangsarchitektur

3 Ein konkreter Fall

Das Kernsystem der Debeka wurde anfangs auf einem GCOS-Großrechner des Herstellers Bull betrieben. In einer ersten Migration wurde die Großrechnerplattform durch AIX-Systeme abgelöst. Die Kernsystemanwendungen wurden 1:1 migriert, und hierfür musste das Verhalten des GCOS-Umfelds auf der neuen Hardware nachgebildet werden. Unter anderem werden COBOL eingesetzt und CODASYL-Datenstrukturen mittels relationaler Datenbanken emuliert. Das aus der GCOS-Welt migrierte Betriebsführungs-konzept basiert auf einer strikten Trennung zwischen Dialog- und Batch-Zeiten. Dies schränkt die Servicequalität erheblich ein und führt zu erhöhten Aufwänden bei der Integration von Softwaresystemen, die nativ im Unix-Umfeld betrieben werden. Die Zielarchitektur sieht daher eine Ablösung des heutigen Technologieumfelds mit COBOL durch den JEE-Technologiestack vor, ergänzt durch Service- und Prozessorientierung sowie unter Verwendung von Web-Standards.

Der Umfang der gesamten Kernanwendung wird am bisherigen Entwicklungsaufwand der COBOL-Programme von mehreren Tausend Personenjahren deutlich. Erschwert wird eine Migration der Anwendungslandschaft durch ständig neue Anforderungen des Gesetzgebers und durch Entwicklungen am Markt. Daher muss die Migration schrittweise und im laufenden Betrieb erfolgen.

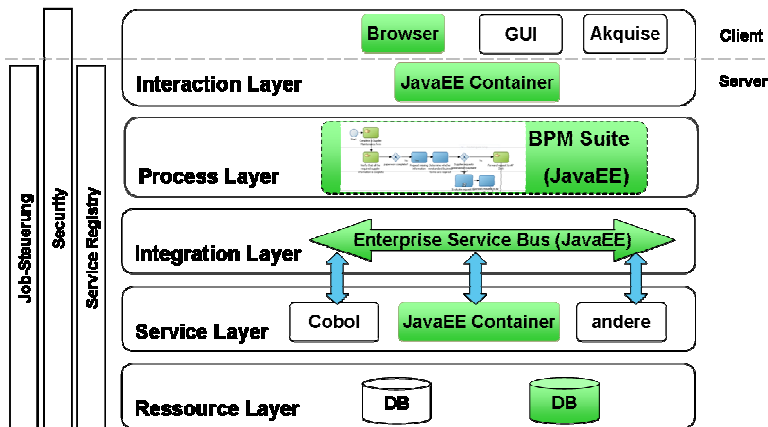


Abbildung 2: Interimsarchitektur

Allerdings können in Java neuentwickelte Anwendungen die existierenden COBOL-Routinen nicht unmittelbar nutzen. Umgekehrt können auch COBOL-Programme Java-Dienste oder externe Webdienste nicht direkt verwenden. Durch diesen Umstand und aufgrund einer starken Abhängigkeit der COBOL-Programme untereinander ist auch die Transformation einzelner Systeme aus der alten in die neue Welt ad hoc nicht möglich. Dies kann erst mittels selbstentwickelter Integrationslösungen gelingen. Während der ersten Migration wurden somit proprietäre Lösungen geschaffen, die immer noch eingesetzt werden. Dies soll bei der kommenden Migration der Anwendungslandschaft vermieden werden.

Die erste Interimsarchitektur sieht daher noch keine Transformation in die neue Welt, sondern die Einführung einer Brückentechnologie vor. Die Brückentechnologie basiert auf Standards und setzt einen ESB ein. Dieser erlaubt die Kombination der neuen Java-Welt mit der alten COBOL-Welt und eröffnet somit die Möglichkeit, ad hoc einzelne Systeme zu migrieren oder neue Java-Anwendungen ins COBOL-Umfeld zu integrieren.

Schon bei der ersten Migration auf Hardwareebene konnten solche Brücken erfolgreich zwischen der GCOS- und der AIX-Welt eingesetzt werden und haben die notwendige 1:1-Migration auf Softwareseite erst ermöglicht.

Das Poster präsentiert anhand ausgewählter Szenarien Details des Idealbildes und der Interimsarchitektur.

Literaturverzeichnis

- [OG11] The Open Group: TOGAF® Version 9.1, an Open Group Standard, 2011.
- [Ha12] Hanschke, I.: Enterprise Architecture Management, Einfach und Effektiv, Carl Hanser Verlag, München 2012.