Effektive Architekturdokumentation mit arc42©

Inhalt

1. Software-Architektur mit arc42©

Die effektive und effiziente Erstellung von Software-Architekturen sowie deren Dokumentation stellt einen der wichtigsten Erfolgsfaktoren von Software-Entwicklungsprojekten dar. Durch Dokumentation von Strukturen innerhalb des Systems können Teams schnell und einfach eingearbeitet werden. Das Betrachten der Systemgrenzen, insbesondere im Zusammenhang mit der Interaktion zu Nachbarsystemen, gewinnt gerade durch die immer schneller wachsenden IT-Landschaften zunehmend an Bedeutung. Das nachfolgende Whitepaper zeigt eine Möglichkeit auf, wie mit UML und einem pragmatischen, iterativen Vorgehen Software-Architekturen beherrschbar gestaltet und dokumentiert werden können.

2. arc42©

arc42© ist eine Kombination aus Prozess und Templates, die es ermöglicht, Software-Architekturen effektiv und pragmatisch zu modellieren und zu dokumentieren. Phillipe Kruchten beschreibt in einem Ansatz, welche Sichten bei der Modellierung software-intensiver Systeme gebildet werden sollten [Kruc 1995]. Das nachfolgende Schaubild (Abbildung 1) zeigt vier Sichten („Views"), die durch Szenarien miteinander verbunden werden .

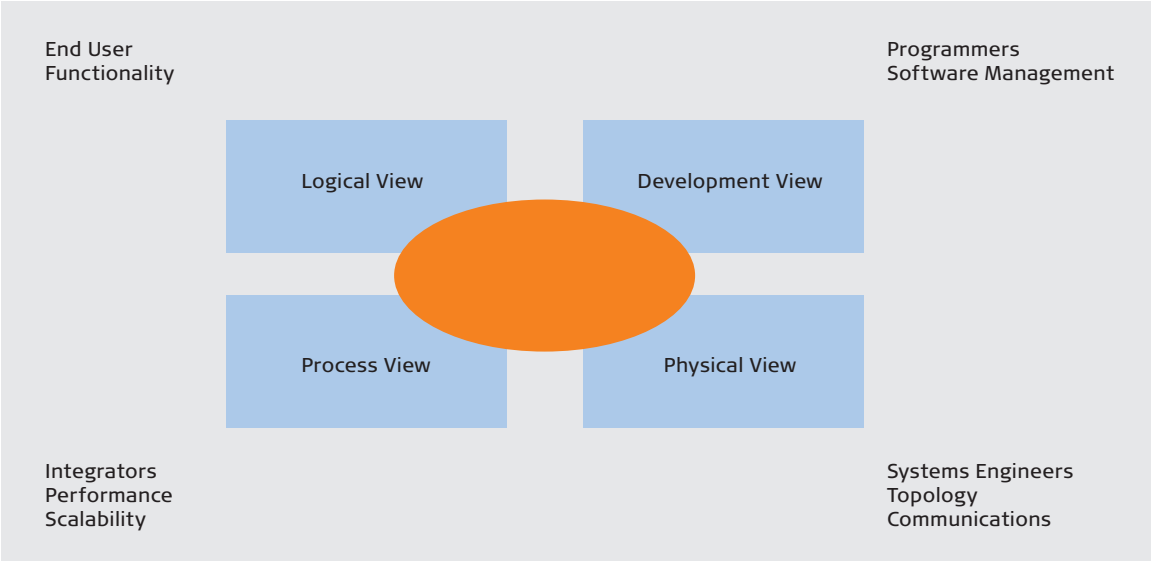

Abbildung 1: 4+1 Architektursichten nach Kruchten©
Quelle: Kruc 1995, Seite 2

In [StHr 2009] wird empfohlen, die Modellierung eines Systems in Kontext-, Baustein-, Verteilungs- und Laufzeitsicht zu strukturieren. Dieser Ansatz wurde maßgeblich durch Kruchten beeinflusst [StHr 2006].

Die folgenden Abschnitte beschreiben, wie UML dazu genutzt werden kann, mit dem von Starke und Hruschka beschriebenen Vorgehen Software-Architektur von Informationssystemen zu modellieren.

3. arc42© -Prozess

Es existieren sechs Kernaufgaben, die Architekten im Rahmen eines Informationssystems (IS) durchführen müssen. Diese Aufgaben werden in Abbildung 2 anhand eines Prozesses veranschaulicht.

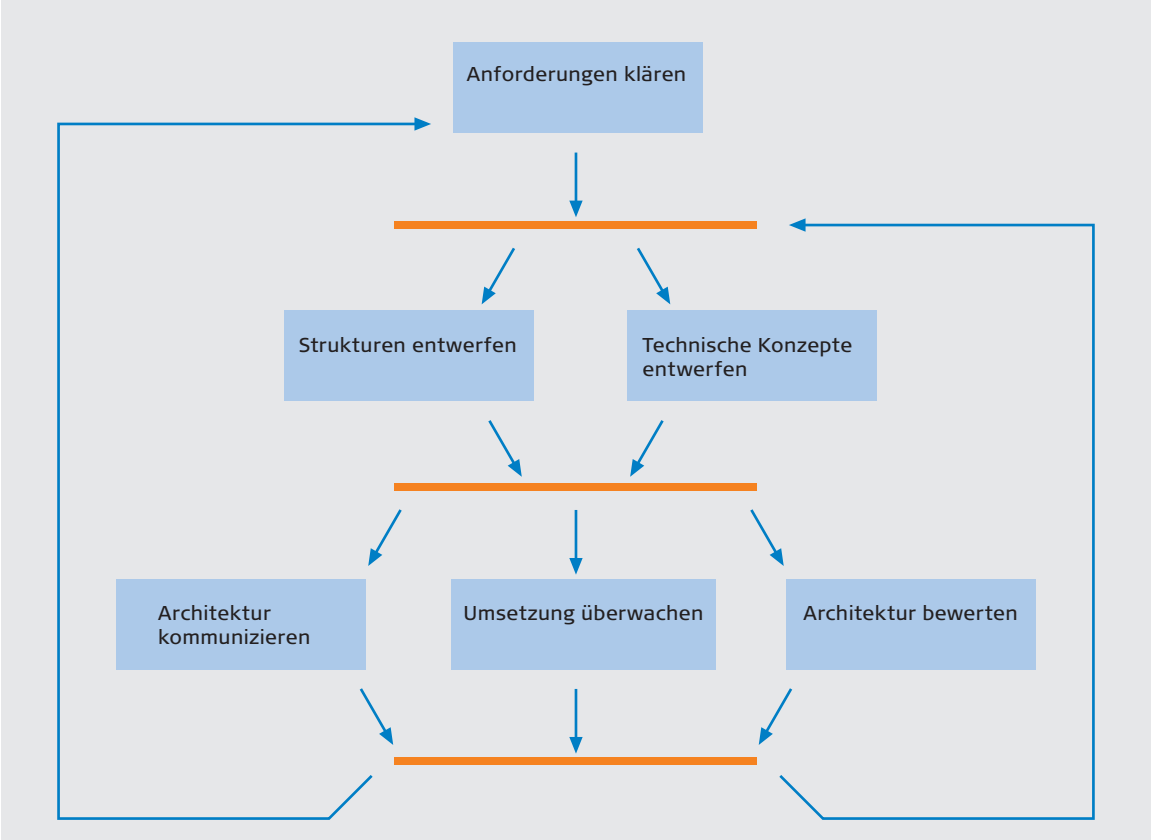

Abbildung 2: Übersicht arc42©-Prozess
Quelle: StHr 2009, Seite 10

Anforderungen und Einflussfaktoren werden vor allem durch Kontextabgrenzung (im fachlichen und technischen Sinne), durch Analyse der Requirements an das IS sowie durch Konkretisierung der Qualitätsziele geklärt. Die Aufgabe „Strukturen entwerfen" bezieht sich auf die angesprochenen Sichten. Hier können pro Sicht unterschiedliche Diagramm-Arten genutzt werden, wie in den nachfolgenden Abschnitten detaillierter beschrieben wird. Technische Konzepte werden in den meisten Fällen direkt an den Bausteinen modelliert, bei denen diese auftreten. Existieren jedoch übergreifende Konzepte wie beispielsweise Persistenz-Mechanismen oder Oberflächen-Technologien, so werden diese an einer zentralen Stelle modelliert. Um sicherzustellen, dass Rahmenbedingungen und Architekturentscheidungen alle Aspekte der ursprünglichen Anforderungen berücksichtigen, wird die Architektur bereits frühzeitig den verschiedenen Stakeholdern des Systems kommuniziert. Während der Implementierung ist darauf zu achten, dass das modellierte System durch die Entwickler entsprechend umgesetzt wird. Ebenfalls muss an dieser Stelle des Prozesses die Bewertung des aktuellen Modells bzw. der aktuellen Architektur durchgeführt werden. Die daraus entstehenden Rückkopplungen fließen dann wiederum in den Strukturentwurf sowie in die Anforderungsklärung mit ein, wodurch eine erneute Iteration durch den Prozess beginnt. [StHr 2009, Seite 10 ff.].

4. Das arc42© Dokumentations-Template

Neben dem Prozess bietet arc42© auch ein detailliertes Dokumentations-Template, das zum Download* zur Verfügung steht. Das vorliegende Whitepaper geht nicht auf alle Gliederungspunkte des Templates ein, sondern lediglich auf die Schwerpunkte: Kontext-, Baustein-, Laufzeit- und Verteilungssicht.Die nachfolgenden Abschnitte zeigen eine exemplarische Modellierung nach arc42© anhand dieses Fallbeispiels:

Bei einer Kreditantragsstrecke können Kunden und Händler Kredite beantragen und erfassen. Nach erfolgreicher Erfassung durch die Anwendung wird über die Kreditprüfung (System) validiert, ob der Kredit genehmigt werden kann oder nicht. Bei Genehmigung wird der Kredit ausbezahlt, ansonsten kommt es zu einem Ablehnungsschreiben.

4.1 Kontextsicht (Kontextabgrenzung)

Die Kontextsicht beschreibt in ihrem Ursprung „das gesamte System als eine einzige Blackbox, mit sämtlichen Schnittstellen" [StHr 2009, Seite 52]. An dieser Stelle ist es sinnvoll, nicht nur die Schnittstellen nach außen, sondern auch die Use-Cases sowie die fachlichen Prozesse, die durch das IS unterstützt werden sollen, aufzuzeigen. Das nachfolgende Diagramm (Abbildung 3) zeigt die Kontext-Sicht mit Hilfe von UML Use-Case-Diagrammen. Zu erkennen ist an dieser Stelle, dass die Antragsstrecke nur die Use-Cases innerhalb der Systeme (Boundary) abdeckt. Die restlichen Use-Cases befinden sich außerhalb der Systemgrenzen und werden durch andere Systeme abgedeckt, mit denen die Antragsstrecke interagiert. Auf Basis dieser Kontext-Sicht werden im nächsten Schritt die Bausteine der Anwendung identifiziert.

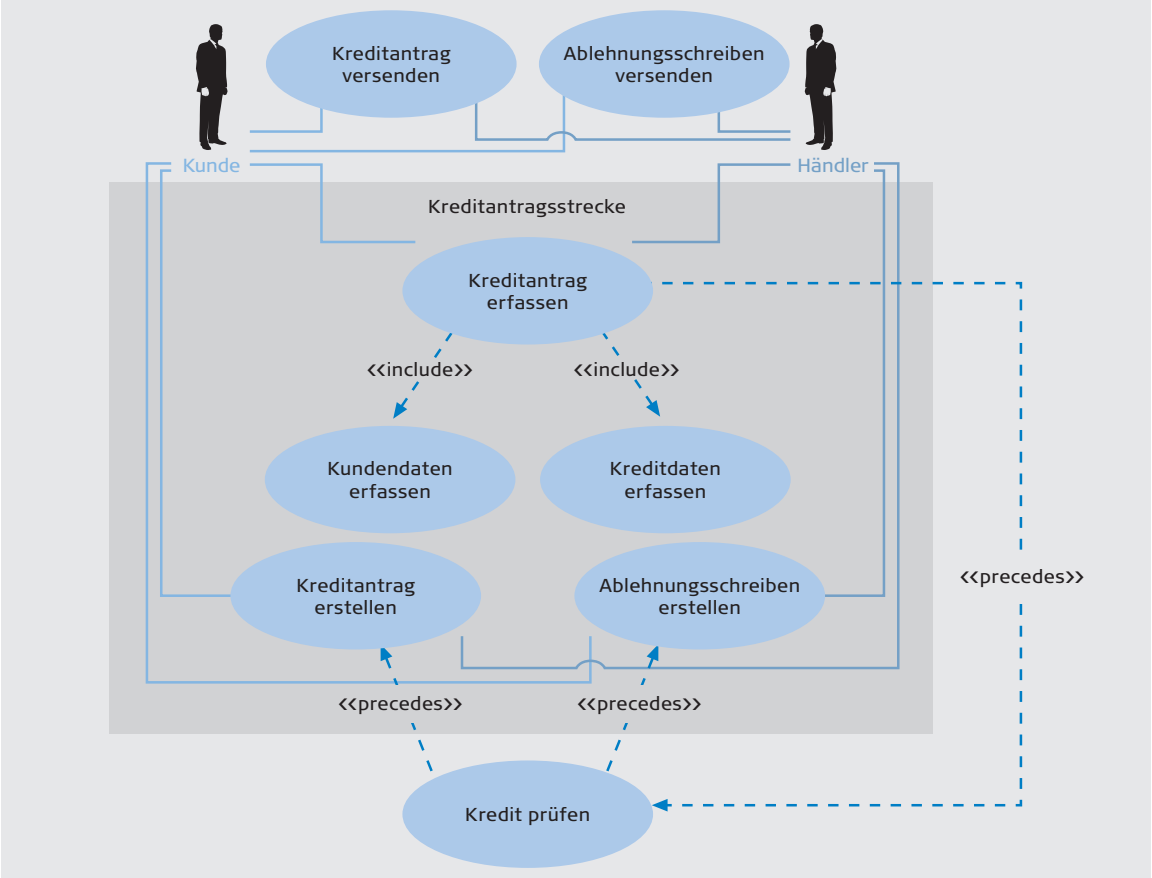

Abbildung 3: Use-Case-Diagramm zur IS-Kreditantragsstrecke

*Siehe http://arc42.de/downloads/downloads.html#arc42_Template_Version_50_DE_De (Stand: 26.06.2011)

4.2 Bausteinsicht

Die Bausteinsicht „zeigt die statische Struktur des Systems, seinen Aufbau aus Softwarebausteinen sowie deren Beziehungen und Schnittstellen untereinander“ [StHr 2009, Seite 28]. Um die Komplexität des Modells zu verringern und es dadurch beherrschbar zu machen, werden an dieser Stelle die einzelnen Bausteine im ersten Schritt als Blackboxes modelliert, da der innere Aufbau der Komponenten, Bausteine und Packages zu Beginn noch nicht relevant ist. Der Fokus liegt hier vielmehr auf der Beziehung zwischen den einzelnen Bausteinen. Zur Identifikation der Bausteine können die Use-Cases der Kontext-Sicht verwendet werden. In diesem Fall werden die Komponenten fachlich geschnitten. Das nachfolgende Komponenten-Diagramm (Abbildung 4) zeigt ein Beispiel für die erste Blackbox-Beschreibung des Systems (Level 0 Blackbox).

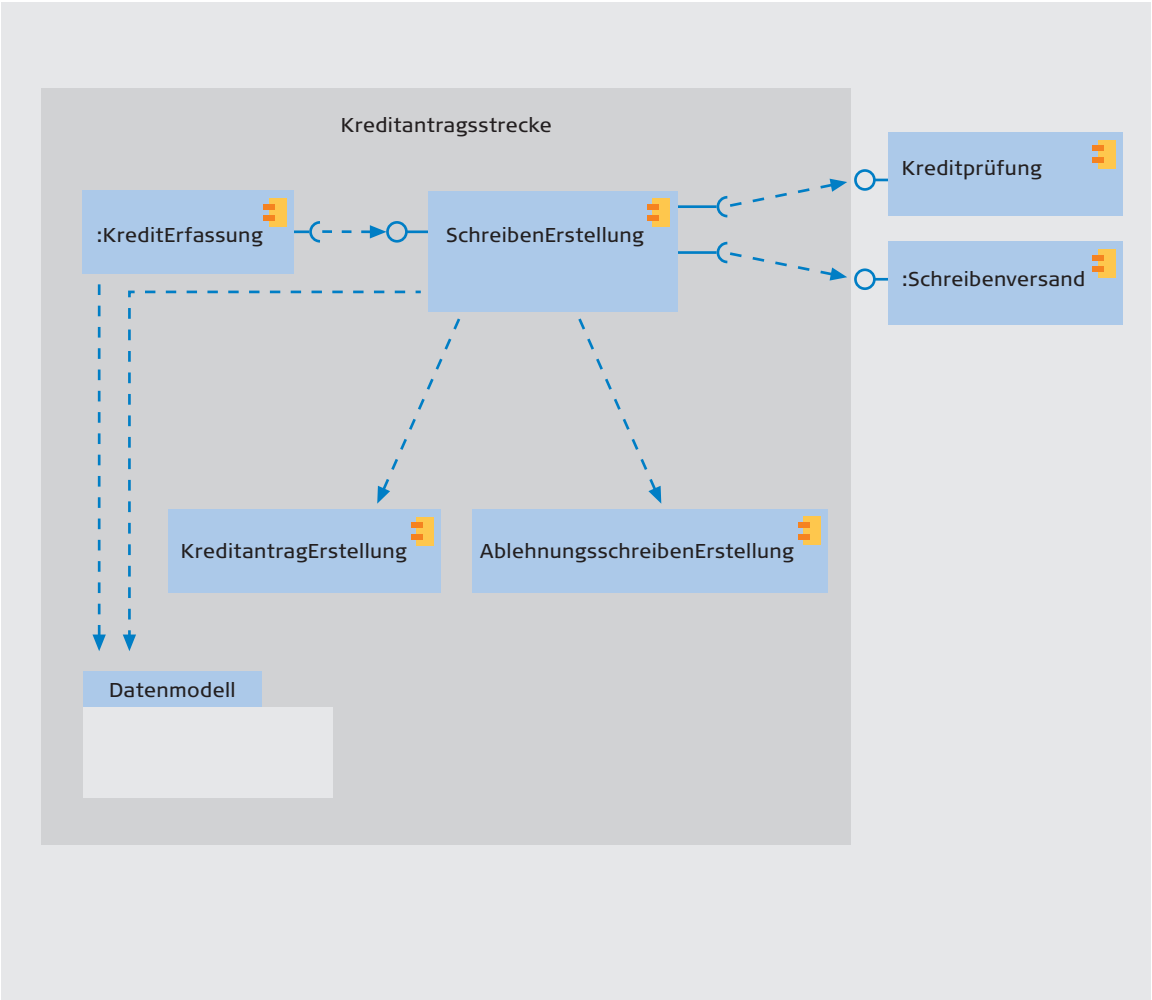

Abbildung 4: Baustein-Sicht Level 0 Blackbox

Danach können die einzelnen Blackbox-Bausteine in Whitebox-Beschreibungen näher spezifiziert werden. Die einzelnen Bestandteile der Whitebox-Beschreibungen sind wiederum selbst Blackboxes, die bei Bedarf immer detaillierter aufgeschlüsselt werden können.

Abbildung 5 zeigt eine weitere Detaillierung des Bausteins Datenmodell (Level 1 Blackbox) in Form eines Klassen-Diagramms.

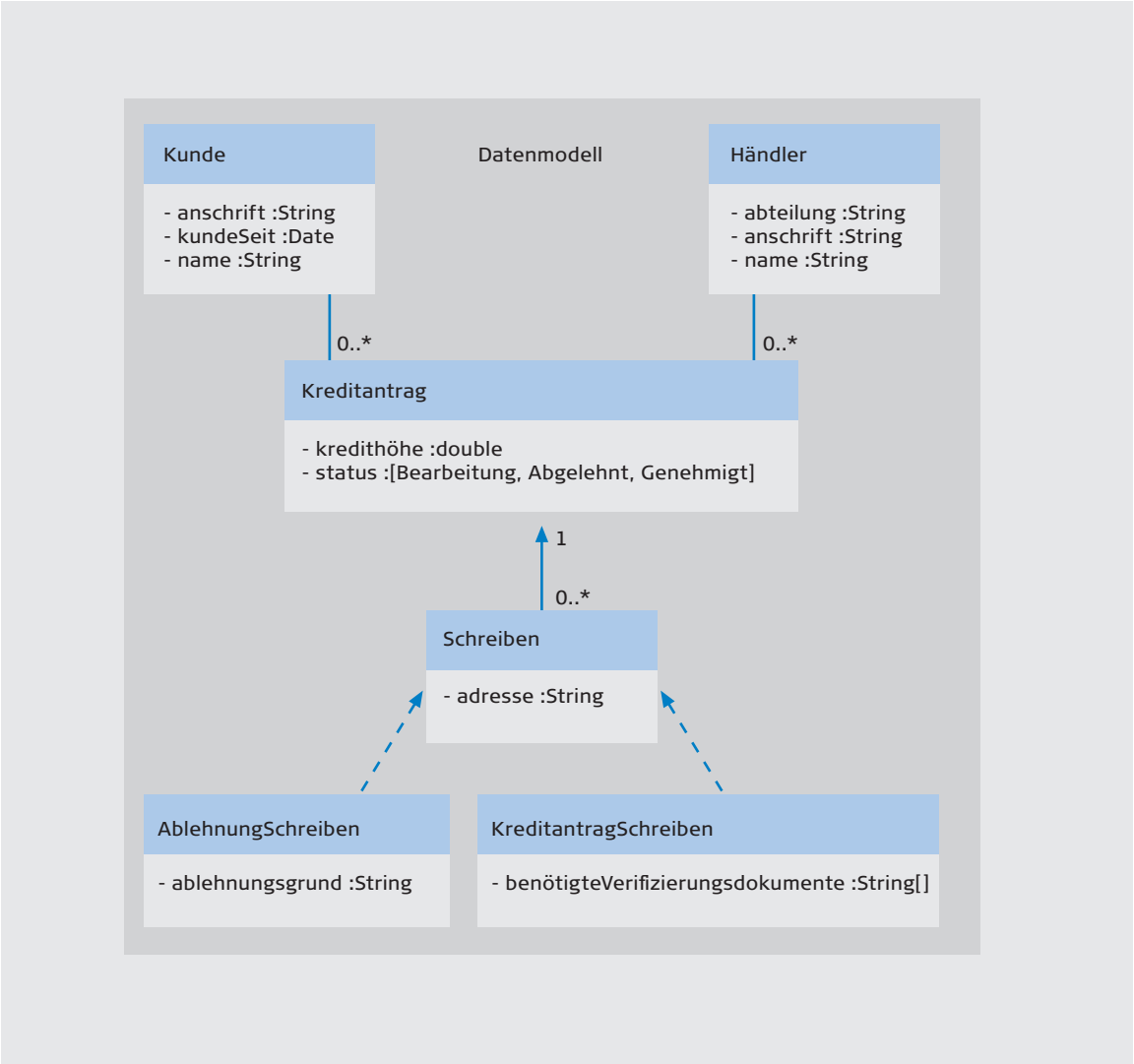

Abbildung 5: Klassendiagramm zum Baustein Datenmodell (Level 1)

Die Attribute der einzelnen Klassen sind nur exemplarisch zu betrachten. Man erkennt, dass eine weitere Verfeinerung dieses Bausteines nicht nötig ist. Komplexere Bausteine können noch detaillierter dargestellt werden. Es besteht keine Notwendigkeit, alle Komponenten eines Systems in gleicher Detaillierungsstufe zu beschreiben.

4.3 Laufzeitsicht

Die Laufzeitsicht zeigt „[...] wie Bausteine des Systems zur Laufzeit miteinander bzw. mit den Nachbarsystemen interagieren." [StHr 2009, Seite 29]. An dieser Stelle sollten jedoch nicht alle Abläufe des Systems dokumentiert werden, sondern nur die wesentlichen. Zur Wahl dieser Szenarien eignen sich die Use-Case-Diagramme der Kontext-Sicht sowie die beteiligten Bausteine und Komponenten aus der Bausteinsicht. Das nachfolgende Diagramm (Abbildung 6) zeigt einen Ablauf zum Szenario „Ablehnungsschreiben versenden".

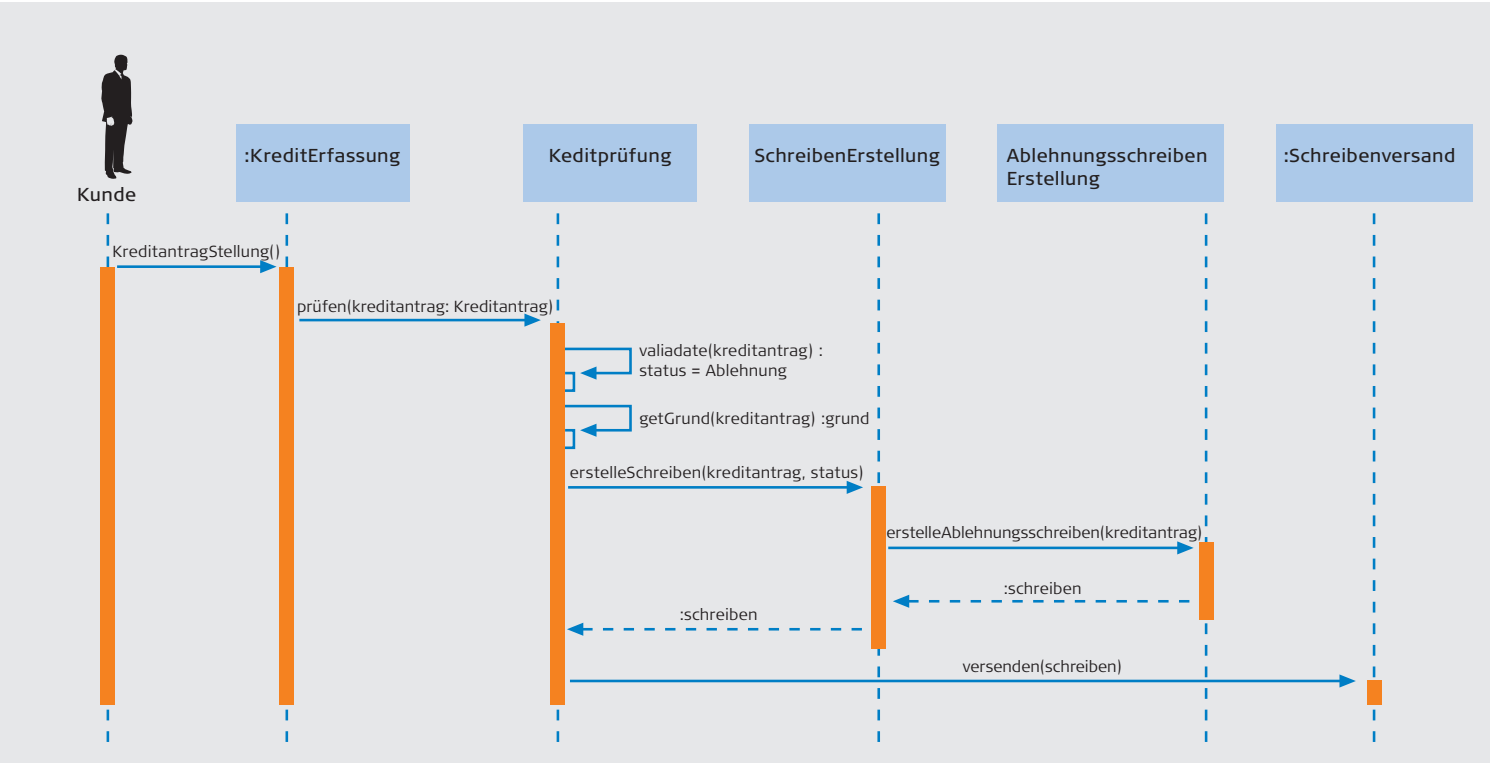

Abbildung 6: Sequenz-Diagramm zum Szenario „Ablehnungsschreiben versenden“

Nachdem der Kunde über die Krediterfassung einen Kreditantrag stellt, wird der Antrag an die Kreditprüfung zur Validierung weitergeleitet. Dort wird ein Status ermittelt und die Komponente „SchreibenErstellung“ aufgerufen. Anhand des Status des Kreditantrages wird erkannt, dass ein Ablehnungsschreiben erstellt werden soll. Daher wird der Aufruf an die entsprechende Komponente zur Erstellung von Ablehnungsschreiben delegiert. Diese erstellt das Schreiben und gibt es zurück. Danach leitet die Kreditprüfungs-Komponente das Schreiben weiter an den Baustein „Schreibenversand“, welcher es druckt und versendet. Die Laufzeitsicht verknüpft somit die beiden Sichten Kontext- und Bausteinsicht über Szenarienmiteinander.

4.4 Verteilungssicht

„Die Verteilungssicht zeigt die Verteilung von Systembestandteilen auf Hard- und Softwareebene.“ [StHr 2009, Seite 29]. In verteilten Systemen ist es wichtig zu modellieren, welche Bausteine der Architektur (siehe Bausteinsicht) auf welchen Prozessoren verteilt vorliegen. Neben der reinen Verteilung werden ebenfalls die Kommunikationswege der einzelnen Bausteine aufgezeigt. Dazu wird das UML Deployment-Diagramm genutzt. Abbildung 7 zeigt ein Deployment-Diagramm für das System „Kreditantragsstrecke“.

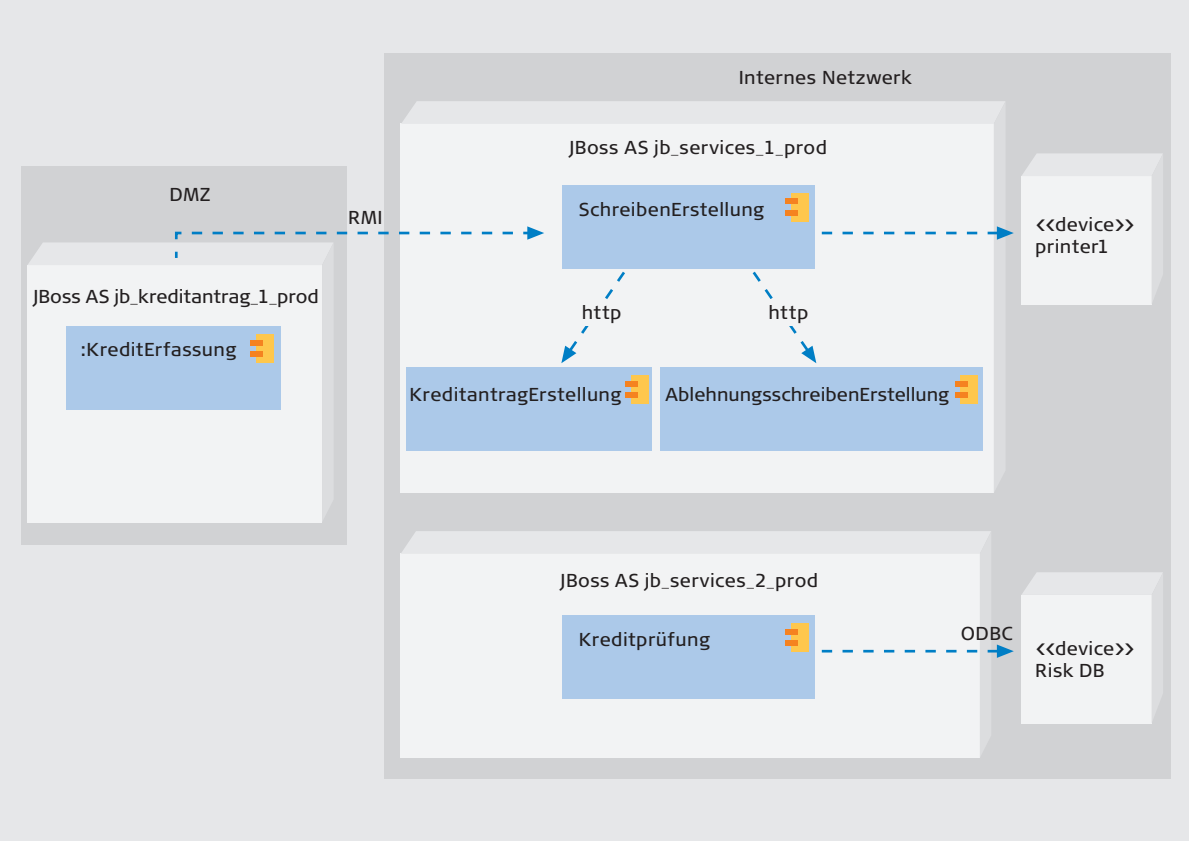

Abbildung 7: Deployment-Diagramm für das IS-Kreditantragsstrecke

Literaturverzeichnis

Kruc 1995 Kruchten, Phillipe: Architectural Blueprints – The „4+1" View Model of Software Architecture, IEEE Software Volume 12, Nr. 6, 1995, S. 42–50

StHr 2006 Dr. Starke, Gernot; Dr. Hruschka, Peter: Praktische Architekturdokumentation: Wie wenig ist genau richtig?, OBJEKTSpektrum, Ausgabe 01/2006, 2006, S. 52–57

StHr 2009 Dr. Starke, Gernot; Dr. Hruschka, Peter: Software-Architektur kompakt – angemessen und zielorientiert. Spektrum Akademischer Verlag, Heidelberg, 2009.

Fragen zu diesen oder anderen Themen beantworten wir Ihnen jederzeit gerne.

Cofinpro AG

Untermainkai 27–28, 60329 Frankfurt am Main
Tel: +49 (0) 69 - 2 99 20 87 60
Mail: welcome@cofinpro.de
www.cofinpro.de

COFINPRO
Finest Processes in Finance

Cofinpro unterstützt Deutschlands führende Finanzdienstleister bei der Verbesserung von Geschäftsprozessen. Wir transformieren Strategien in Prozesse und implementieren diese Prozesse in der IT. Durch Business Transformation schaffen wir für unsere Kunden entscheidende Wettbewerbsvorteile in einem Finanzmarkt, der sich immer noch sehr schnell verändert und vor spannenden Herausforderungen steht. Unser Erfolgskonzept ist dabei ein ganz besonderes: Wir kombinieren hohe Fach- und Technologiekompetenz mit den Methoden unseres Business Engineering Frameworks – durch diese einzigartige Kompetenzbündelung erreichen wir für unsere Kunden exzellente Ergebnisse: „Finest Processes in Finance".

Cofinpro – die Experten für Kredit und Wertpapier