



Vortrag:

Historisch gewachsen? Java-Architekturen angemessen dokumentieren

Stefan Zörner, oose Innovative Informatik GmbH, Hamburg

**BerlinExpertDays (BED-Con) in Berlin,
Freie Universität, Campus Dahlem, 7. April 2011**



Zusammenfassung

Historisch gewachsen?

Java-Architekturen angemessen dokumentieren



Architekturdokumentation wird oft als lästige Pflicht angesehen. Dabei ermöglicht das angemessene Festhalten die Kommunikation Ihrer Konzepte im Team und dem Auftraggeber gegenüber überhaupt erst. Anhand konkreter Beispiele aus der Java-Welt erfahren Sie, welche Tools, Notationen und Arbeitsergebnisse sich in der Praxis bewähren. Häufige Herausforderungen werden ebenso diskutiert wie typische Werkzeugketten. Wiki oder UML-Tool? Oder was dazwischen? Und wie kommt man falls verlangt jederzeit zu einer druckbaren Dokumentation? Diese Session unterstützt Sie dabei, Ziele, Entscheidungen, Richtlinien und Konzepte im Team zu verankern, anstatt sie zu vergessen.

Stefan Zörner, Stationen

- 1991-94 Ausbildung Math.-techn. Assistent bei der **Bayer AG**
- **Studium** Mathematik (Diplom 1998), Schwerpunkt Informatik
- 1998-2001 **Mummert + Partner AG**, Berater, u.a. Sun-Trainer
- 2001-2006 **IBM** e-business Innovation Center, IT-Architekt
- Seit Juli 2006: Berater und Trainer bei **oose** in Hamburg
 - Schwerpunkt: Softwareentwurf und Java-Technologien
 - Stefan.Zoerner@oose.de



Veröffentlichungen, Vorträge

- **Bücher** „Portlets“, 2006
„LDAP für Java-Entwickler“, 3. Auflage 2007
- Artikel u.a. in Java Magazin und bei IBM developerWorks
- Vorträge bei JAX und W-JAX seit 2002, Advisory Board



Dies und das

- Seit 2005 Mitarbeit im **Apache** Directory Project, szoerner@apache.org
- iSAQB Certified Professional for Software Architecture
- OMG Certified UML Professional (Intermediate)
- SpringSource Certified Spring Professional



Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3** Die Aufgabe
- 4** Sichten
- 5** Lochen und abheften
- 6** Schluss und Aus(-blick)

Agenda

- 1** Motivation – Montag Morgen
- 2 Softwarearchitektur
- 3 Die Aufgabe
- 4 Sichten
- 5 Lochen und abheften
- 6 Schluss und Aus(-blick)

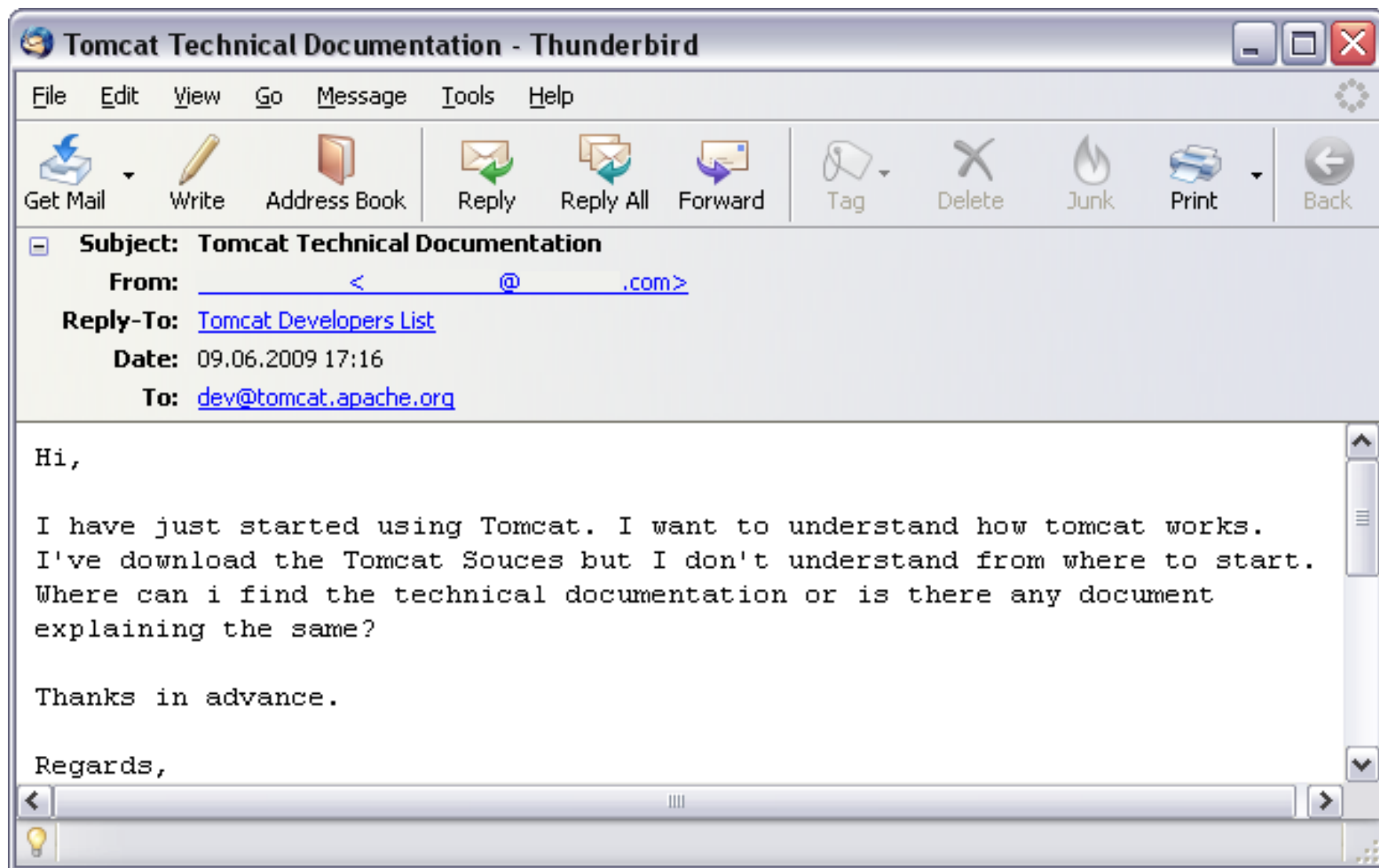
Montag Morgen ...

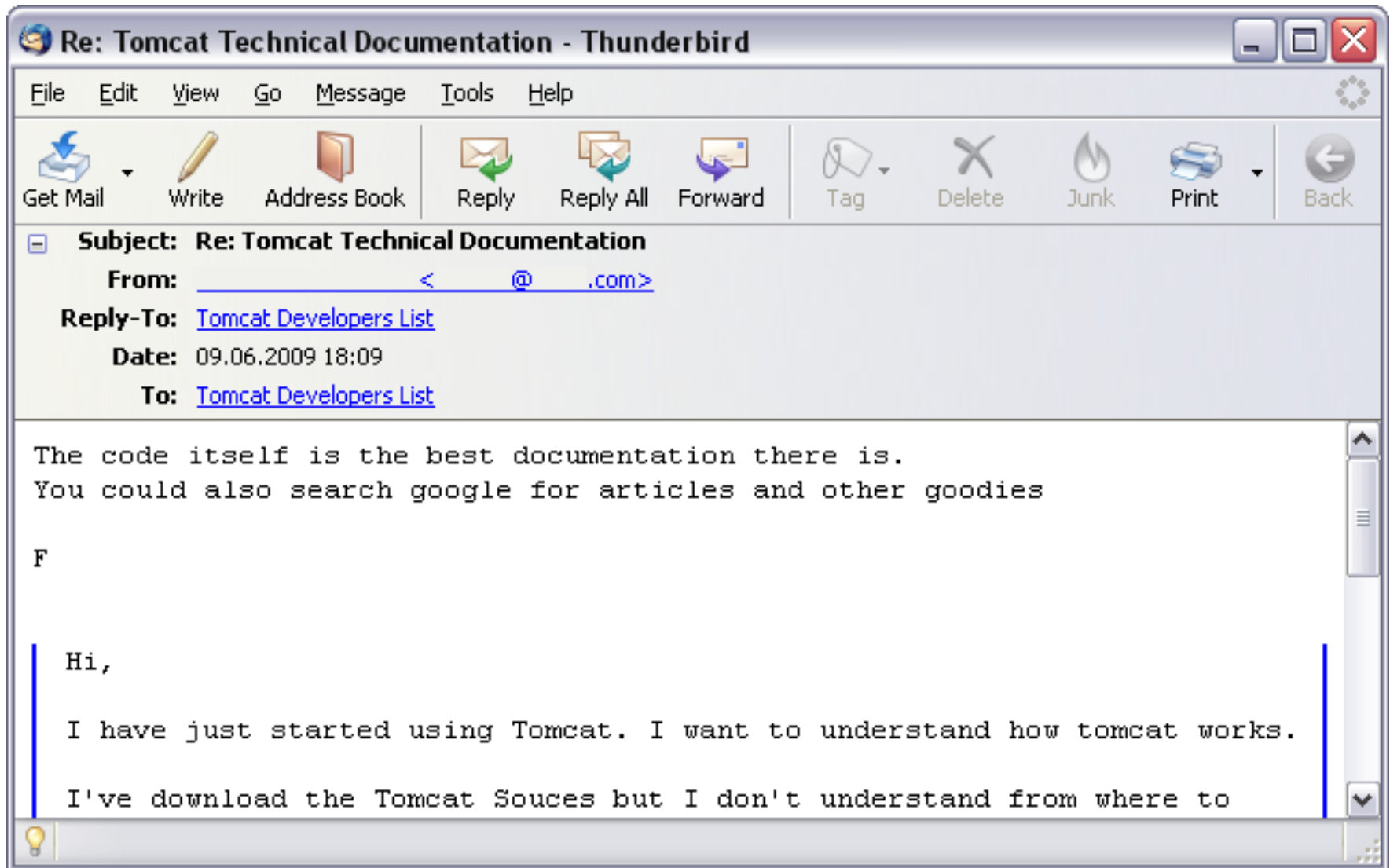
April 2011						
«						»
Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Fragen, die neue Mitarbeiter so stellen ...

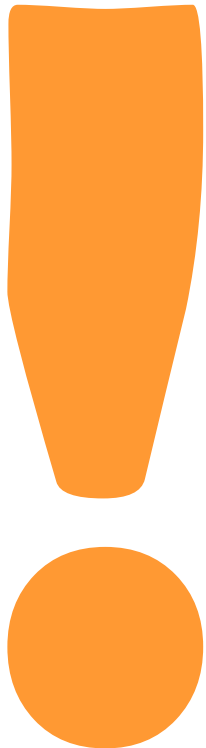


- Wie checke ich die Sourcen aus, und wie baue ich die Software?
- Warum sind bei mir Tests rot?
- Was brauche ich für Tools?
- Wenn ich neue Funktionalität hinzufügen soll – wie stelle ich das an? Hier ist doch schon was Ähnliches, kann ich das wiederverwenden?
- Was leistet das System überhaupt?
- Aus welchen Bestandteilen besteht die Software?
- Wie arbeiten diese zusammen?
- Ist das irgendwo beschrieben?
- Warum benutzt ihr noch JDK 1.4?
- Wieso habt Ihr das denn so gemacht?
- ...





Antworten, die neue Mitarbeiter daraufhin erhalten



- Steht alles im Wiki.
- Das haben wir nicht dokumentiert – wir gehen agil vor.
- Das war schon so, als ich neu war.
- Das ist historisch gewachsen.

„Dokumentation“ als Fremdwort

Do|ku|men|ta|ti|on [...zion] [lat.] die; -, -en: 1. a) Zusammenstellung u. Ordnung von Dokumenten und Materialien jeder Art, durch die das Benutzen und Auswerten ermöglicht oder erleichtert wird ...



Dokumentation zählt nicht zu den Lieblingsbeschäftigungen von Entwicklern. Es wird oft als lästige Pflicht angesehen. Dabei ermöglicht das angemessene Festhalten die Kommunikation Ihrer Konzepte im Team und dem Auftraggeber gegenüber überhaupt erst.

Architekturdokumentation – Ziele



Architekturdokumentation will dabei unterstützen ...

- **die Umsetzung zu leiten**
- **eine Konsistente Lösung zu erreichen und zu erhalten**
- **die Lösung nachvollziehbar zu machen**

Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3 Die Aufgabe
- 4 Sichten
- 5 Lochen und abheften
- 6 Schluss und Aus(-blick)

Definitionen zu Softwarearchitektur

- Es gibt nicht die eine allgemein akzeptierte Definition für Softwarearchitektur
- Das Software Engineering Institute (SEI) sammelt sogar Definitionen:

The screenshot shows a Mozilla Firefox browser window displaying the SEI website. The page title is "Software Architecture for Software-Intensive Systems". The URL in the address bar is "http://www.sei.cmu.edu/architecture/definitions.html". The page features the SEI logo and the Carnegie Mellon University name. A navigation menu includes "HOME", "CONTACT US", "SITE MAP", "SEARCH", and a "GO" button. Below the menu, there are tabs for "About the SEI", "Areas of Work", "Work with Us", "Products & Services", and "Publications". The main content area is titled "Software Architecture for Software-Intensive Systems" and "How Do You Define Software Architecture?". It includes a quote by Eoin Woods: "Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled." Below the quote, it states: "The following sections attempt to capture an appropriate cross section of what is meant by software architecture." There is also a small image of a computer keyboard.

→ <http://www.sei.cmu.edu/architecture/definitions.html>

Architekturentscheidungen ...



“Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.” (Eoin Woods)

- Architekturentscheidungen sind diejenigen, die sich im weiteren Verlauf nur sehr schwer revidieren lassen.
- Konsequenzen: höhere Kosten, Zeitverlust, ggf. scheitert das Vorhaben

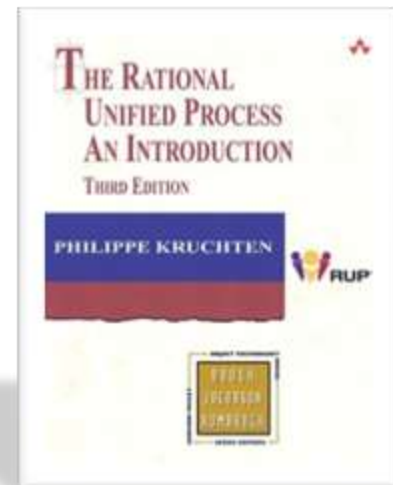
Eine (!) konkrete Definition

Architektur := Σ wichtige Entscheidungen

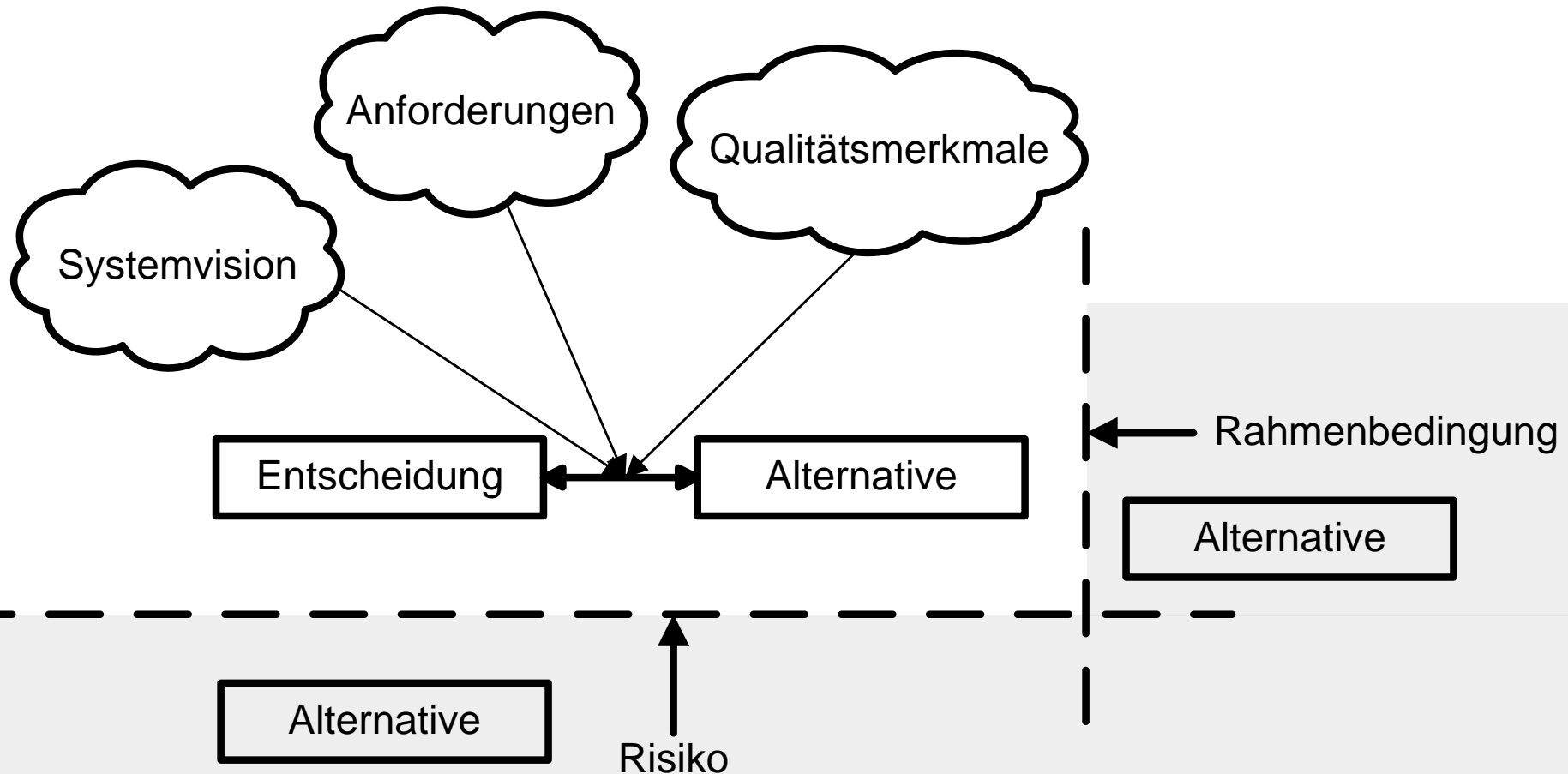
Softwarearchitektur umfasst die Summe verschiedener wichtiger Entscheidungen über

- die Auswahl von Strukturelementen und deren Schnittstellen, aus denen das System zusammengesetzt ist
- das Verhalten und Zusammenspiel dieser Elemente
- den hierarchischen Aufbau von Subsystemen
- den zugrunde liegenden Architekturstil
- ...

G. Booch, P. Krutchen, K. Bittner and R. Reitman.
The Rational Unified Process — An Introduction. 1999.

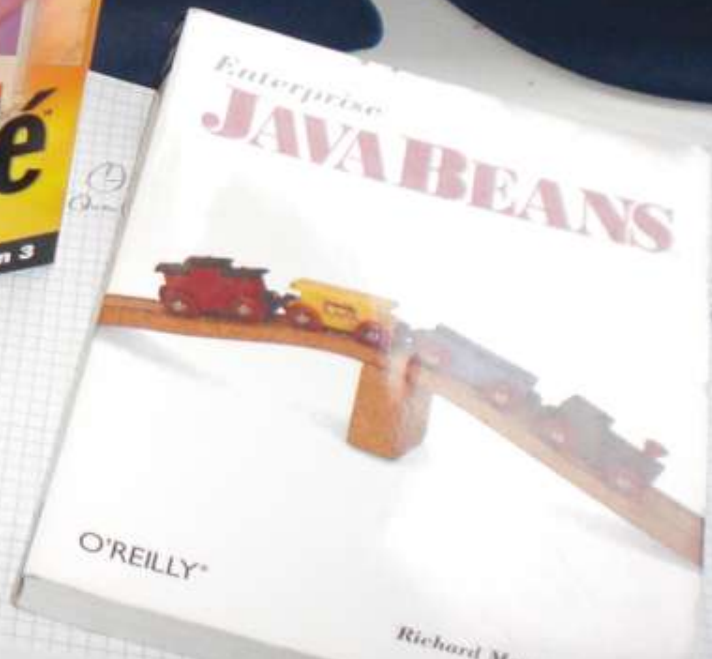


Einflussfaktoren auf Architekturentscheidungen



Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3** Die Aufgabe
- 4 Sichten
- 5 Lochen und abheften
- 6 Schluss und Aus(-blick)



Homepage ActiveMQ

Apache ActiveMQ -- Index - Mozilla Firefox

File Edit View History Bookmarks Tools Help


http://activemq.apache.org/

ActiveMQ

The Apache Software Foundation
http://www.apache.org/

Apache ActiveMQ > Index Download | JavaDocs More... | Source | Forums | Support

Download it Today!



Apache ActiveMQ is the most popular and powerful open source Message Broker and [Enterprise Integration Patterns](#) provider.

Apache ActiveMQ is fast, supports many [Cross Language Clients and Protocols](#), comes with easy to use [Enterprise Integration Patterns](#) and many [advanced features](#) while fully supporting [JMS 1.1](#) and [J2EE 1.4](#). Apache ActiveMQ is released under the [Apache 2.0 License](#)

Grab yourself a [Download](#), try our [Getting Started Guide](#), surf our [FAQ](#) or start [Contributing](#) and join us on our [Discussion Forums](#).

Features

- Supports a variety of [Cross Language Clients and Protocols](#) from Java, C, C++, C#, Ruby, Perl, Python, PHP
 - [OpenWire](#) for high performance clients in Java, C, C++, C#
 - [Stomp](#) support so that clients can be written easily in C, Ruby, Perl, Python, PHP, ActionScript/Flash, Smalltalk to talk to ActiveMQ as well as any other [popular Message Broker](#)

Overview

- Index
- News
- New Features
- Getting Started
- FAQ
- Articles
- Download

Search

Sub Projects

- Camel
- CMS
- NMS

Community

- Support
- Contributing
- Discussion Forums
- Mailing Lists
- IRC

Architekturziele als Produktkarton



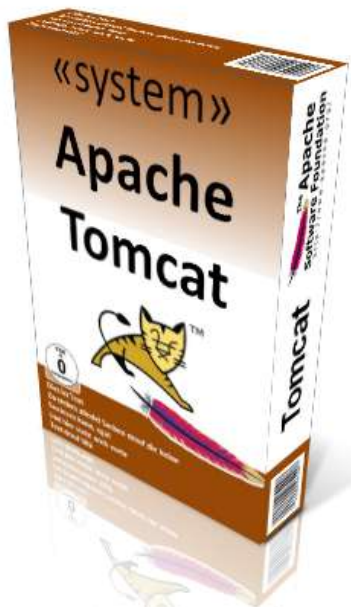
- Was entwickeln wir eigentlich?
- Was sind die zentralen Features des Systems?
- Wem nützt es?
- Wie unterscheidet es sich von Produkten der Mitbewerber?

Speziell für die Architektur

- Welche Qualitätsmerkmale (= Ziele) sind besonders wichtig?
- Welche Rahmenbedingungen sind interessant?

Es ist eine Ihrer Aufgaben als Softwarearchitekt, die Idee des Systems im Entwicklungsteamteam zu verankern.

Fallbeispiel: Ein Produktkarton für Apache Tomcat



- Apache Tomcat ist die beliebteste Laufzeitumgebung für Java Webapplikationen aller Art (Webshops, Communities, Web Services ...).
- Als Open Source Referenzimplementierung der fundamentalen Java EE Bestandteile Java Servlets und JavaServer Pages setzt Tomcat Standards.
- Produktreife, Zuverlässigkeit, Administrierbarkeit und Clusteringfähigkeit machen Tomcat zur idealen Plattform für den Betrieb unternehmenskritischer Anwendungen weltweit

Wesentliche Features

- Läuft auf allen *nix-Plattformen, Microsoft Windows, Mac OS, ...
- Unterstützt Applikationen mit JSF, Spring, GWT ...
- Integration mit nativen Web-Servern (IIS, Apache, ...), Datenbanken, LDAP ...

Virtuelle Produktkarton erstellen ...



How to Create a Product Box in Photoshop: 9 steps (with pictures) - wikiHow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.wikihow.com/Create-a-Product-Box-in-Photoshop

wikiHow
The How-to Manual That You Can Edit

Home > Categories > Computers and Electronics > Software > Graphics > Adobe Graphics Programs Write an Article

Related wikiHows

- Replace Text in Adobe Photoshop
- Create a Simple Glass Button in Photoshop
- Create a Magazine in Photoshop

Ads by Google

Photoshop 6.0
Free Photoshop Tutorials
Free Money
Download Photoshop
How to Make Money Online

Ads by Google

Free Photoshop Tutorials

CS4 videos Watch an expert show you Photoshop!

Article Discuss Edit History Share

How to Create a Product Box in Photoshop

Do you want to design an attractive box for your product? With Photoshop, you can create one, just like this:

z.B.

→ <http://www.wikihow.com/Create-a-Product-Box-in-Photoshop>

Kontext – Software agiert nicht allein ...

- Stets gibt es Beteiligte außerhalb des Systems:
 - Anwendergruppen, die Funktionalität nutzen und erwarten
 - Fremdsysteme, die zur Ausführung erforderlich sind

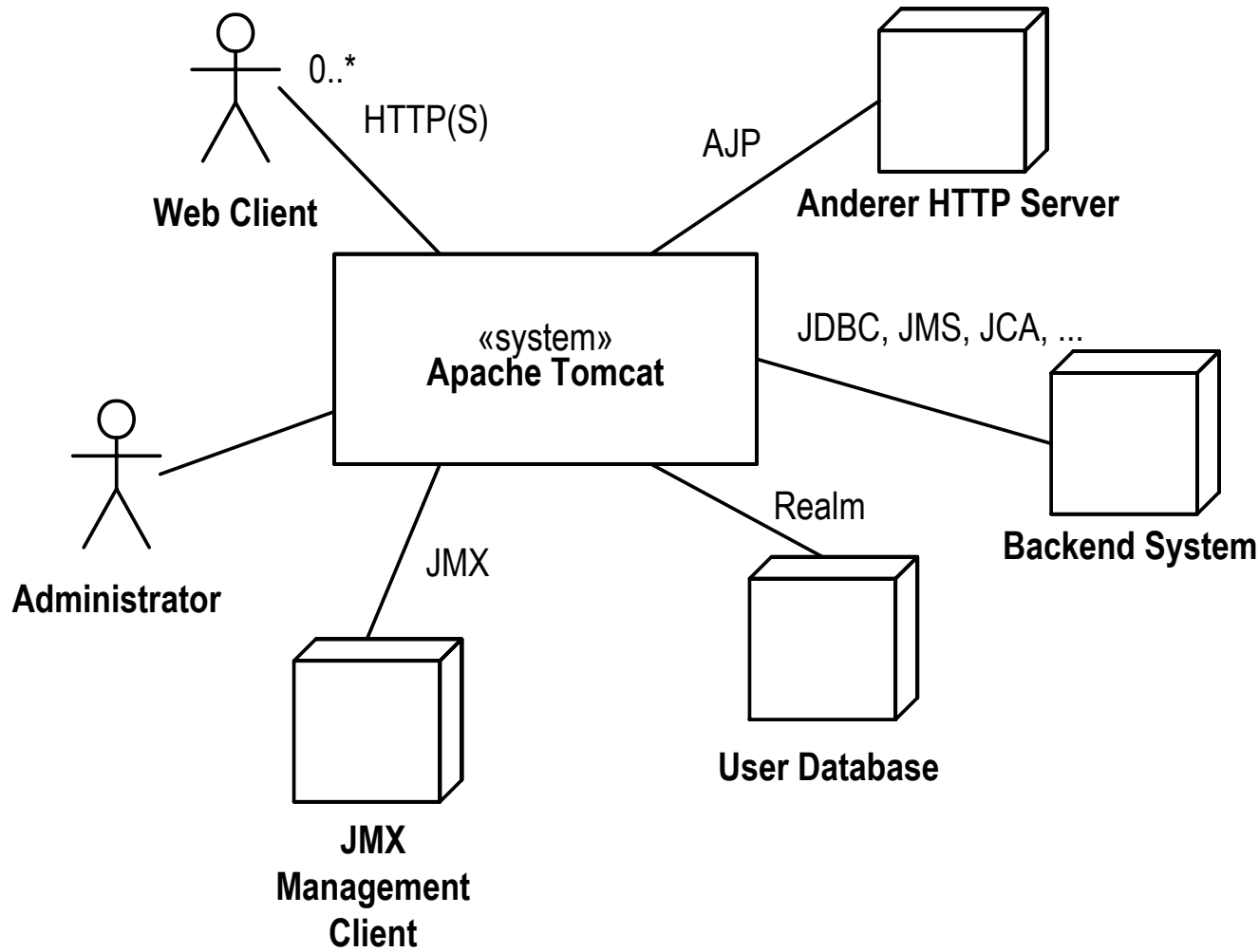


Die **Kontextsicht** zeigt das Umfeld, d.h. alle außerhalb des eigenen Systems liegenden Akteure, mit denen direkt kommuniziert wird.

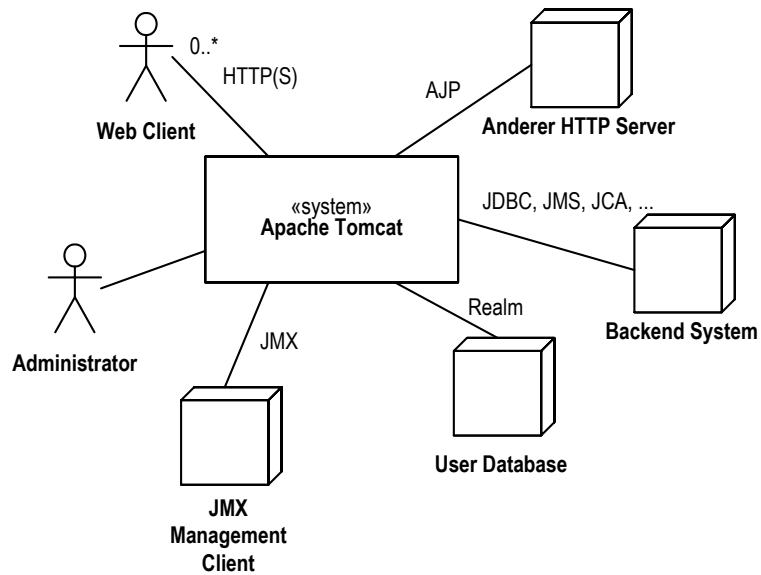
Systemkontextdiagramm: Visualisierung des Umfelds

- das zu beschreibende System im Mittelpunkt als Blackbox
- drum herum die direkt beteiligten Benutzer und Fremdsysteme
- Verbindung zwischen einem solchen Akteur und dem System drückt Interaktion aus.

Eine Kontextsicht für Apache Tomcat (UML).



Systemkontext und Architekturziele



Was ist drum herum?



Was steckt drin?

Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3** Die Aufgabe
- 4** Sichten
- 5 Lochen und abheften
- 6 Schluss und Aus(-blick)

Schwanensee (1877)

Scene

from "Swan Lake"

P.Tschaikowsky / Trans. H.M.

Moderato *mf* *espress.* *p*

Beispiel Tanznotation



Fig 1

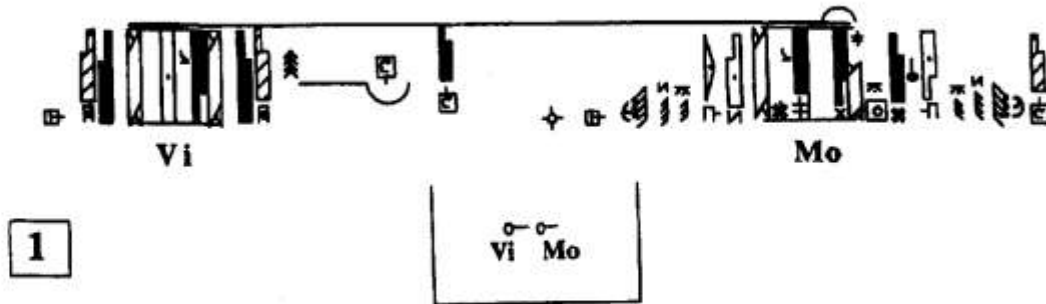
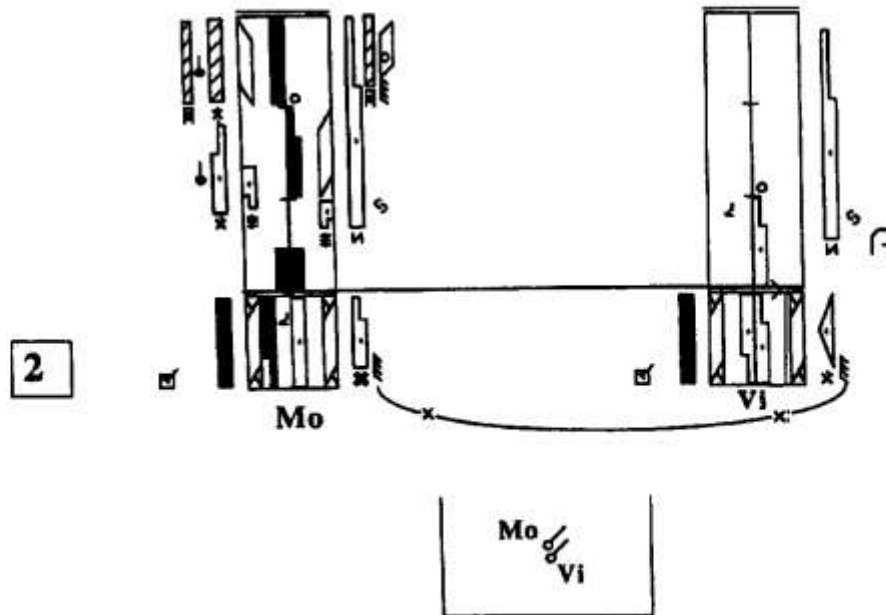


Fig 2



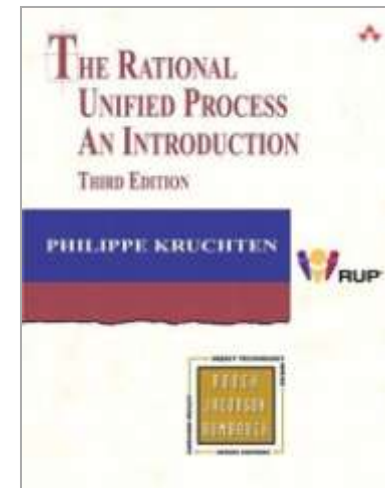
Analogie zur Softwarearchitektur: Views (Sichten)

- Es ist sinnvoll, bestimmte Aspekte einer Software mit Bilder statt textuell zu beschreiben
- Ein einzelnes Bild reicht in der Regel nicht aus
 - ➔ Unterschiedliche Sichten für unterschiedliche Aspekte



Beispiel: Rational Unified Process (P. Kruchten)

- 4 + 1 Views:
 - Logical View
 - Development View
 - Process View
 - Physical View
 - Scenarios



Literaturtipp zu dem Thema:



Effektive Software-Architekturen Ein praktischer Leitfaden

von Gernot Starke

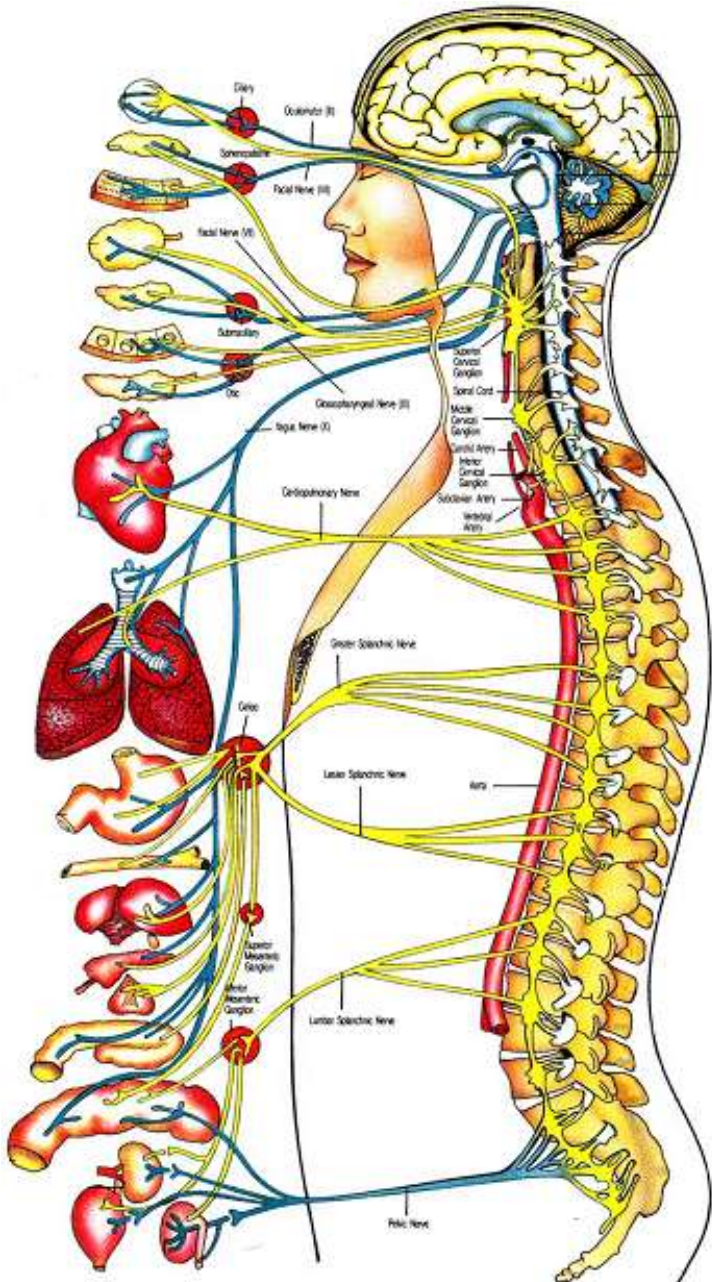
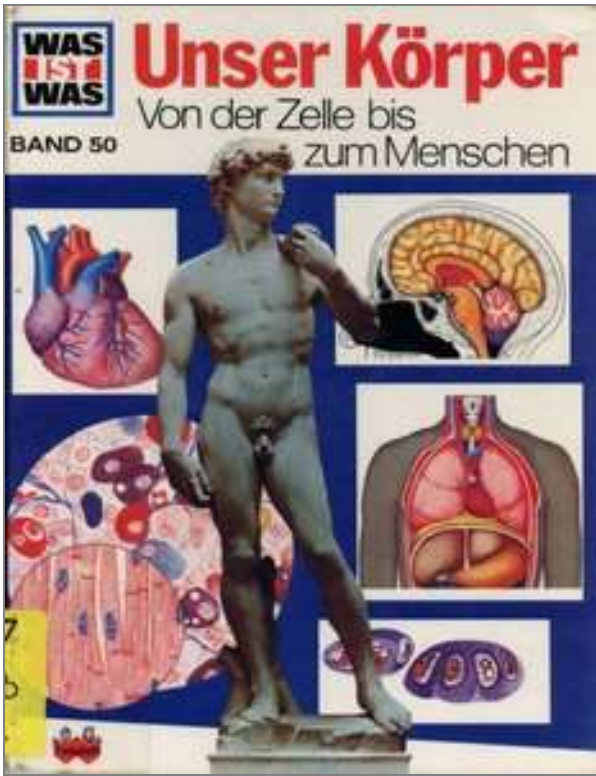
449 Seiten,

Hanser Fachbuch; 4. Auflage (2009)

ISBN 978-3446420083

Dort beschriebene Sichten (u.a.)

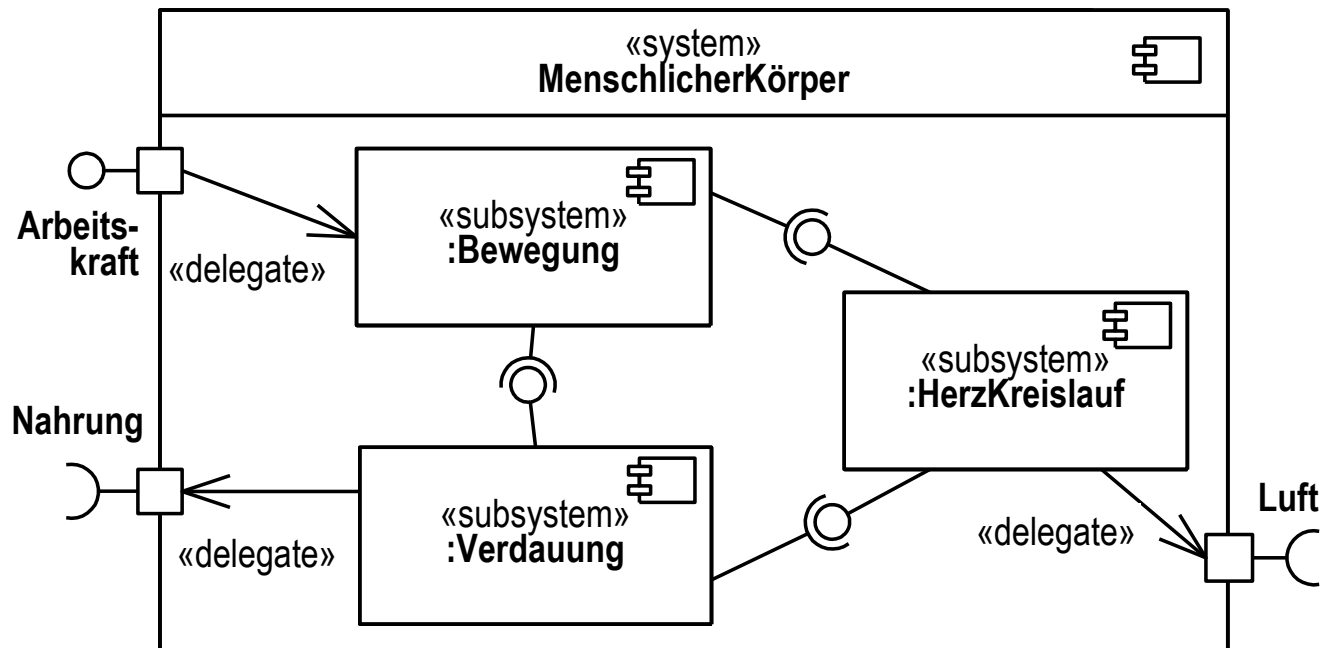
- Kontextsicht
- Bausteinsicht (= Struktur)
- Laufzeitsicht (= Verhalten, Dynamik)
- Verteilungssicht



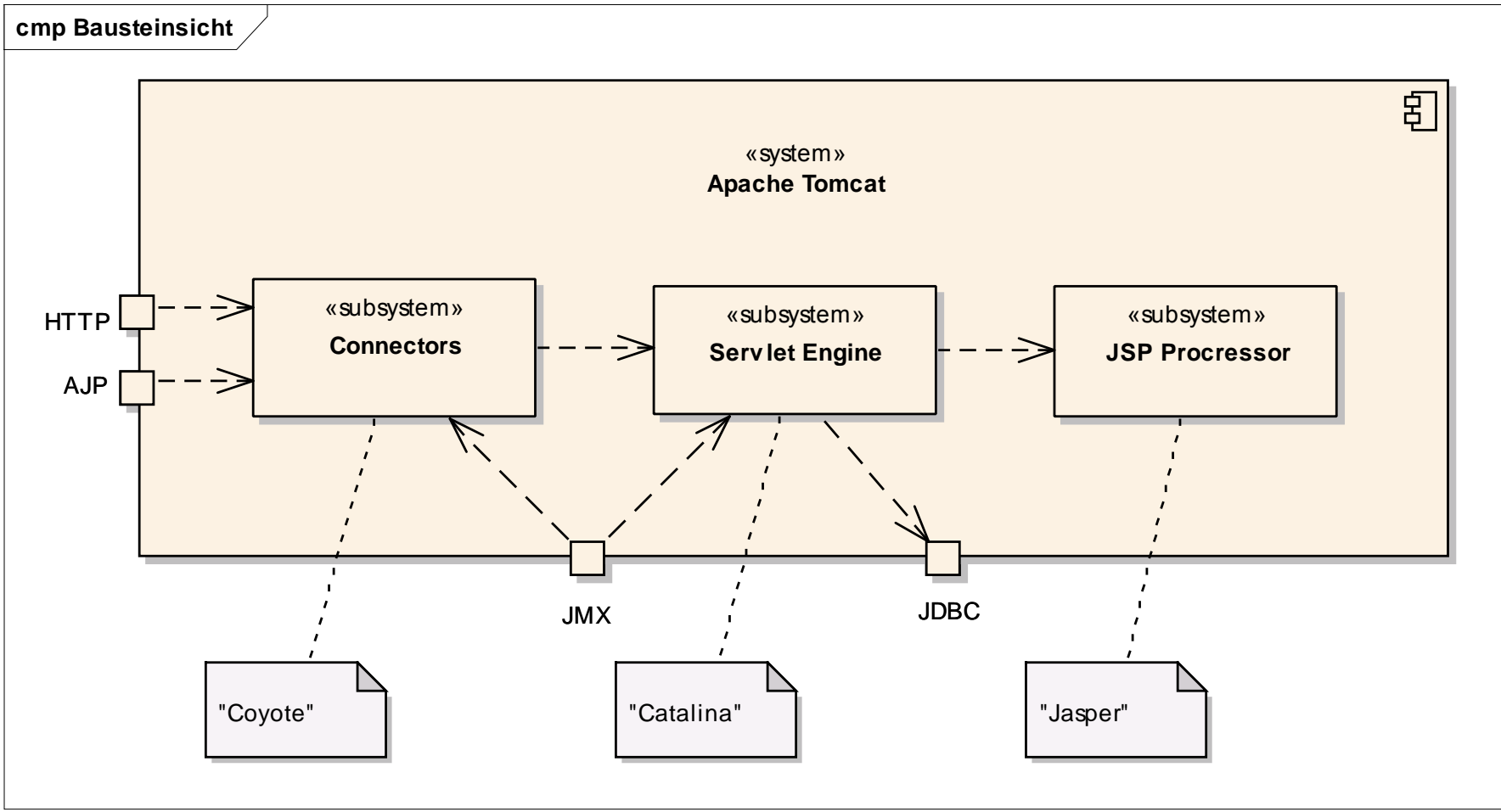
Die Bausteinsicht

„Die Bausteinsicht bildet die Funktionalität des Systems auf Software- oder Implementierungsbausteine ab. Die Sicht macht Struktur und Zusammenhänge zwischen den Bausteinen der Struktur explizit “ (G. Starke)

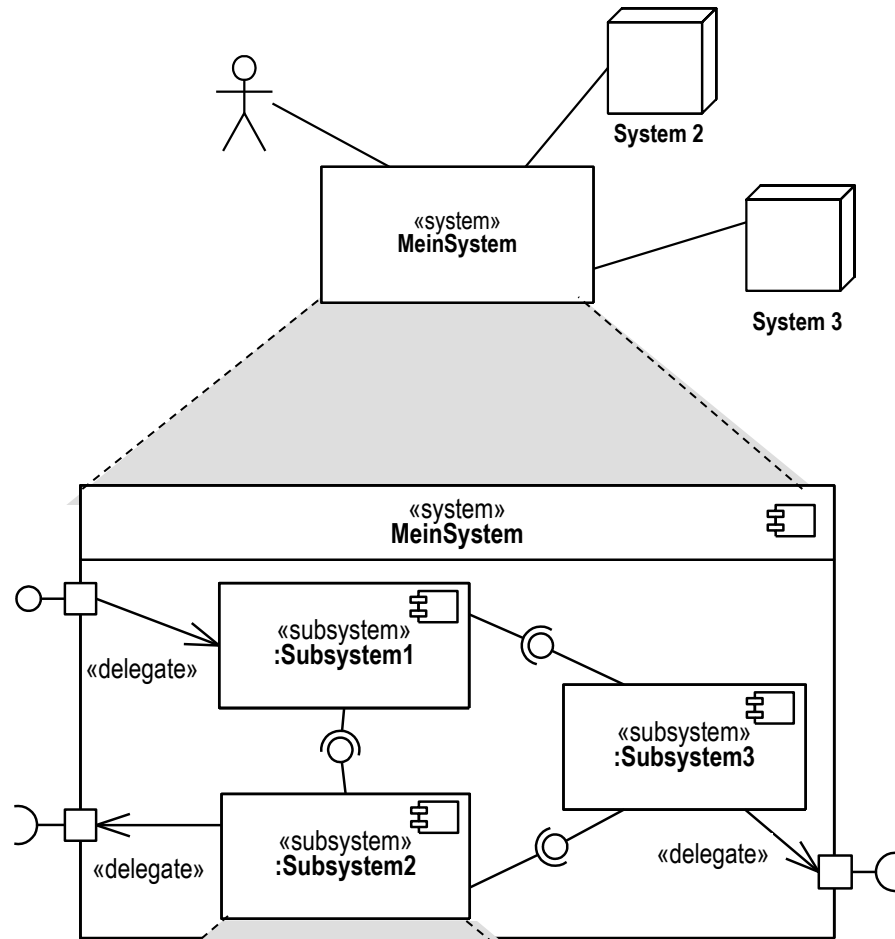
Beispiel (UML, Kompositionsstrukturdiagramm)



Apache Tomcat: Bausteinsicht



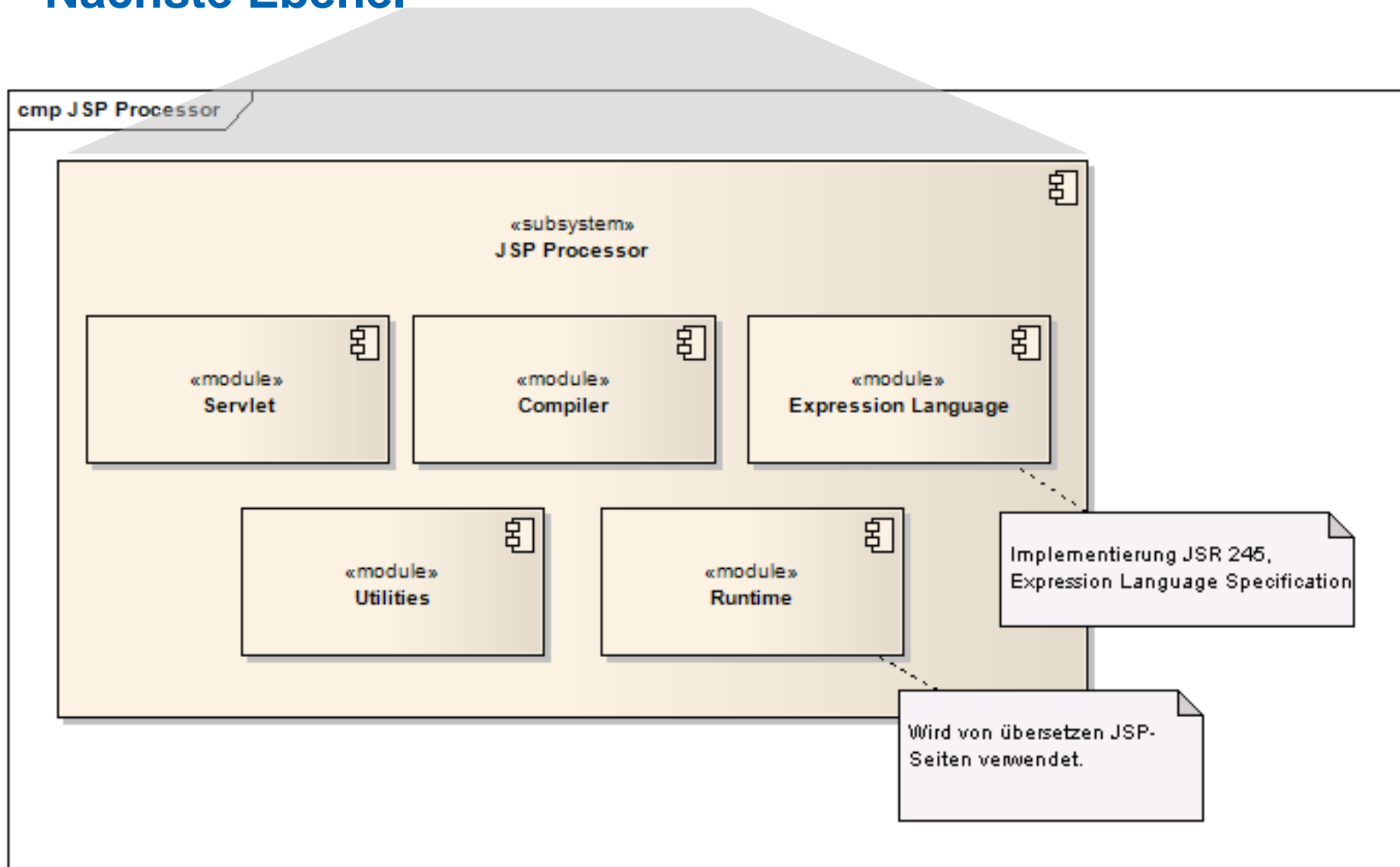
Zusammenspiel Kontextsicht / Bausteinsicht



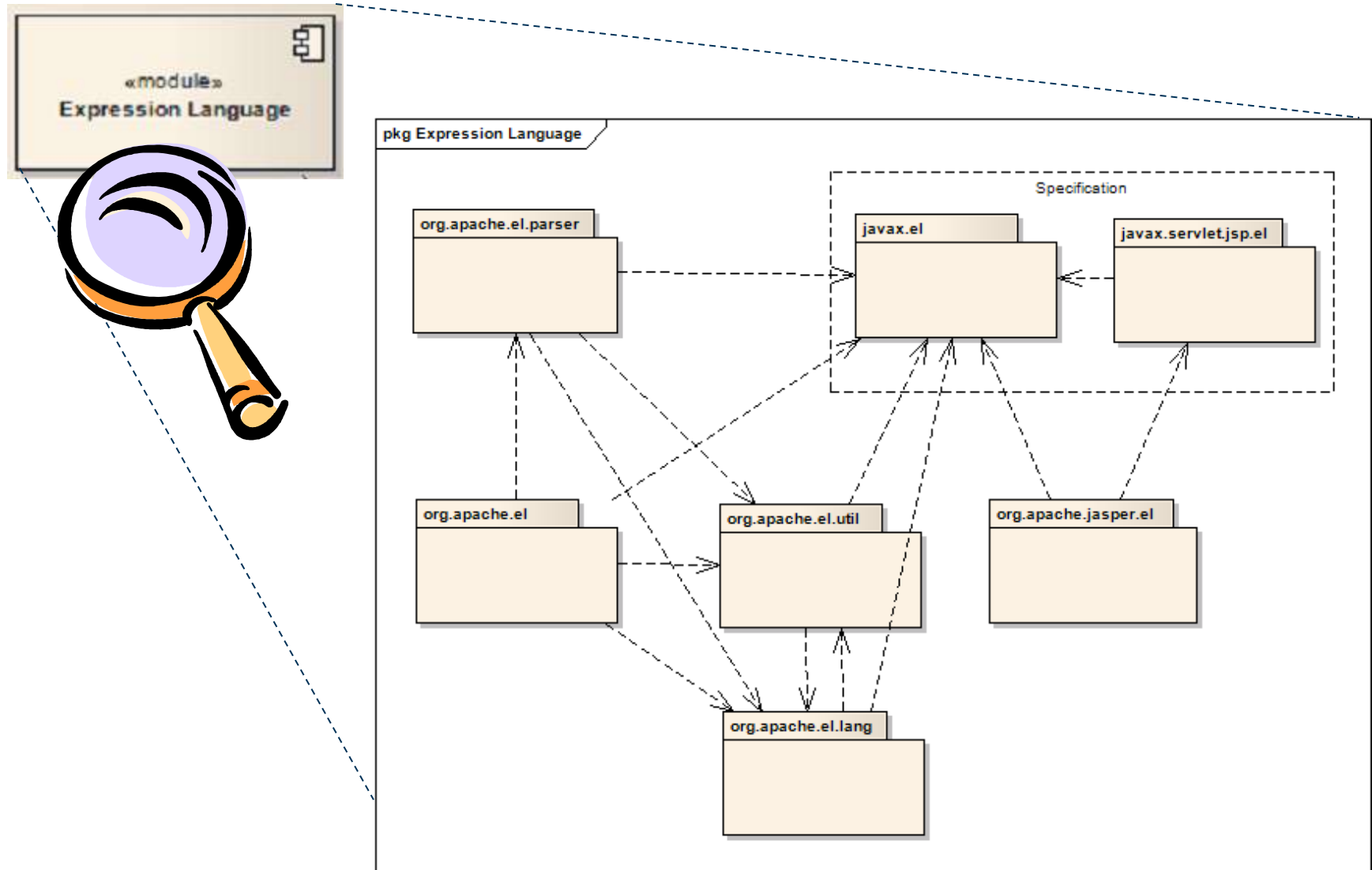
Blackbox

Whitebox

Nächste Ebene.



Nächste Ebene ...



Die Laufzeitsicht – In Bewegung



- Die Bausteinsicht bietet lediglich eine statische Sicht
- Oft bringt erst die Zusammenschau mit dynamischen Aspekten Einsichten, wie das System eigentlich funktioniert, bzw. zu verwenden oder zu erweitern ist.

Die **Laufzeitsicht** (alternativ: Verhaltenssicht) beschreibt, wie Softwareelemente zur Laufzeit interagieren, bzw. wie ein Element selbst sich verhält.

Laufzeitsicht und UML

- Die UML bietet verschiedene Modellelemente und Diagrammtypen für die Laufzeitsicht an, z.B.
 - Aktivitätsdiagramm
 - Sequenzdiagramm
 - Zustandsdiagramm

Beispiel: Apache Tomcat

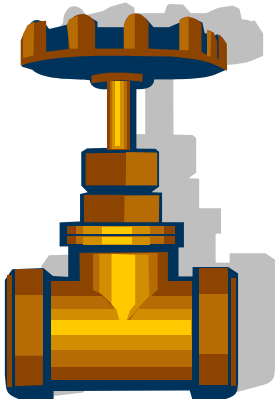
Implementierung einer eigenen Tomcat-Komponente

„Ein Design sollte offen für Erweiterungen, aber geschlossen für Änderungen sein.“ (Open Closed Principle)

Bertrand Meyer 1988



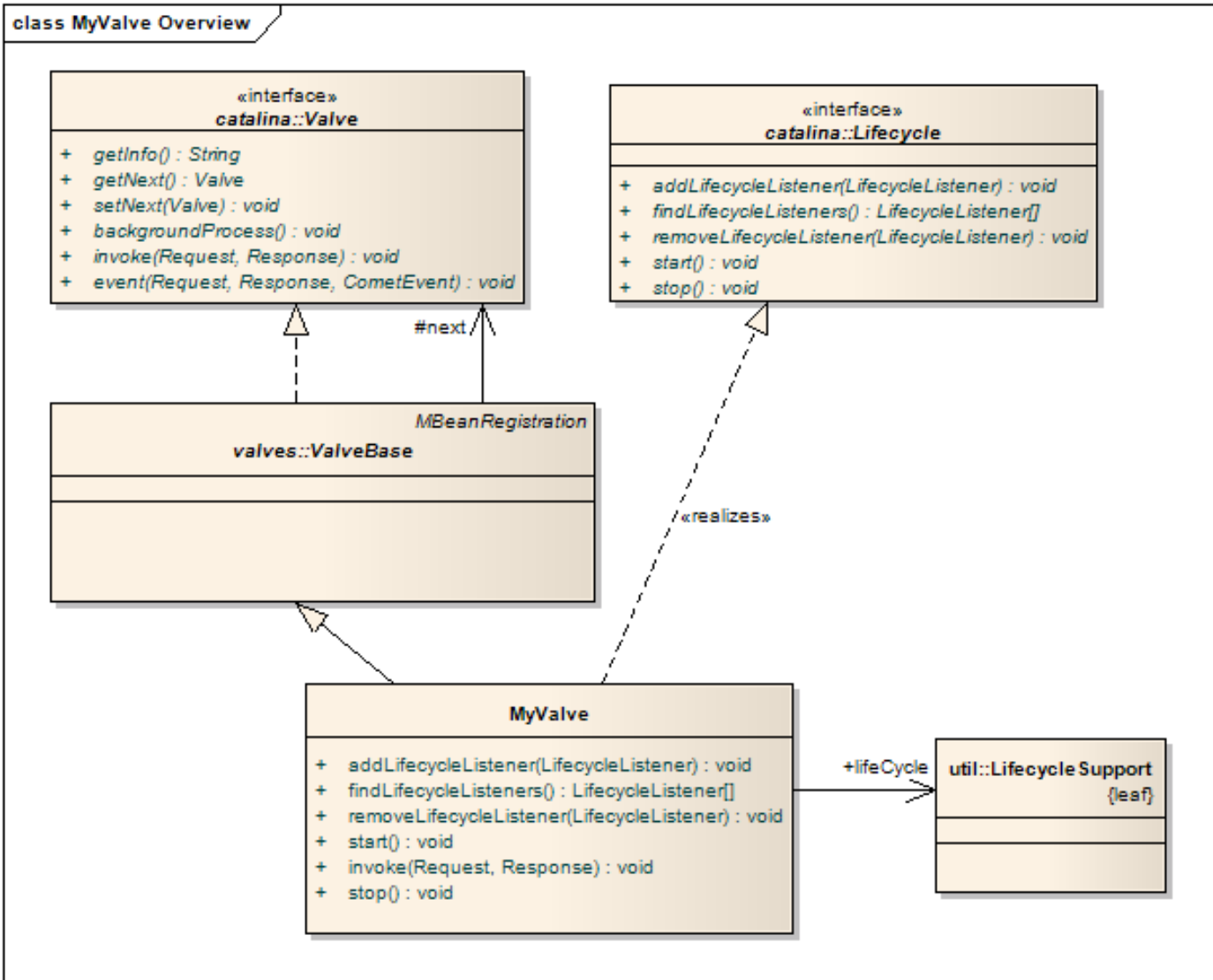
- Tomcat kennt verschiedene Abstraktionen, die gewollte Erweiterungspunkte darstellen (z.B. Connector, Realm)
- Frage: Wie dokumentiert man die Implementierung von Erweiterungen?



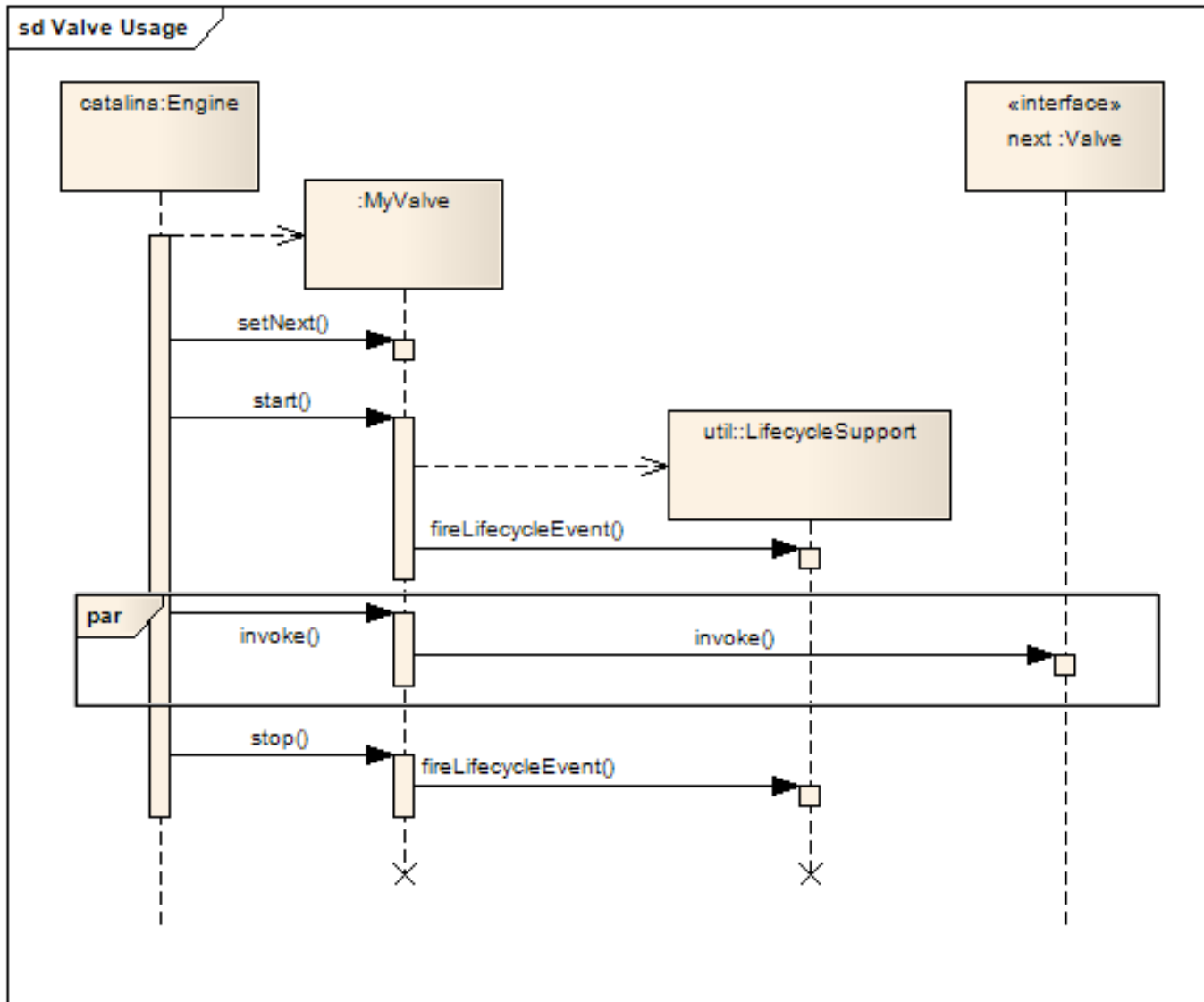
Beispiel: Valve

- Ein Valve (dt. "Ventil") ist eine Anfragen verarbeitende Komponente, die mit einem Container assoziiert ist.
- Üblicherweise bilden eine Kette von Valves eine Pipeline (d.h. ein Valve kennt seinen Nachfolger).

Statische Sicht.



Dynamische Sicht.



Die Verteilungssicht – Ja wo laufen sie denn?



- Die bisherigen Sichten blenden Betriebsaspekte völlig aus.
- Wie verteilt sich die Lösung auf z.B. auf unterschiedliche Rechner?

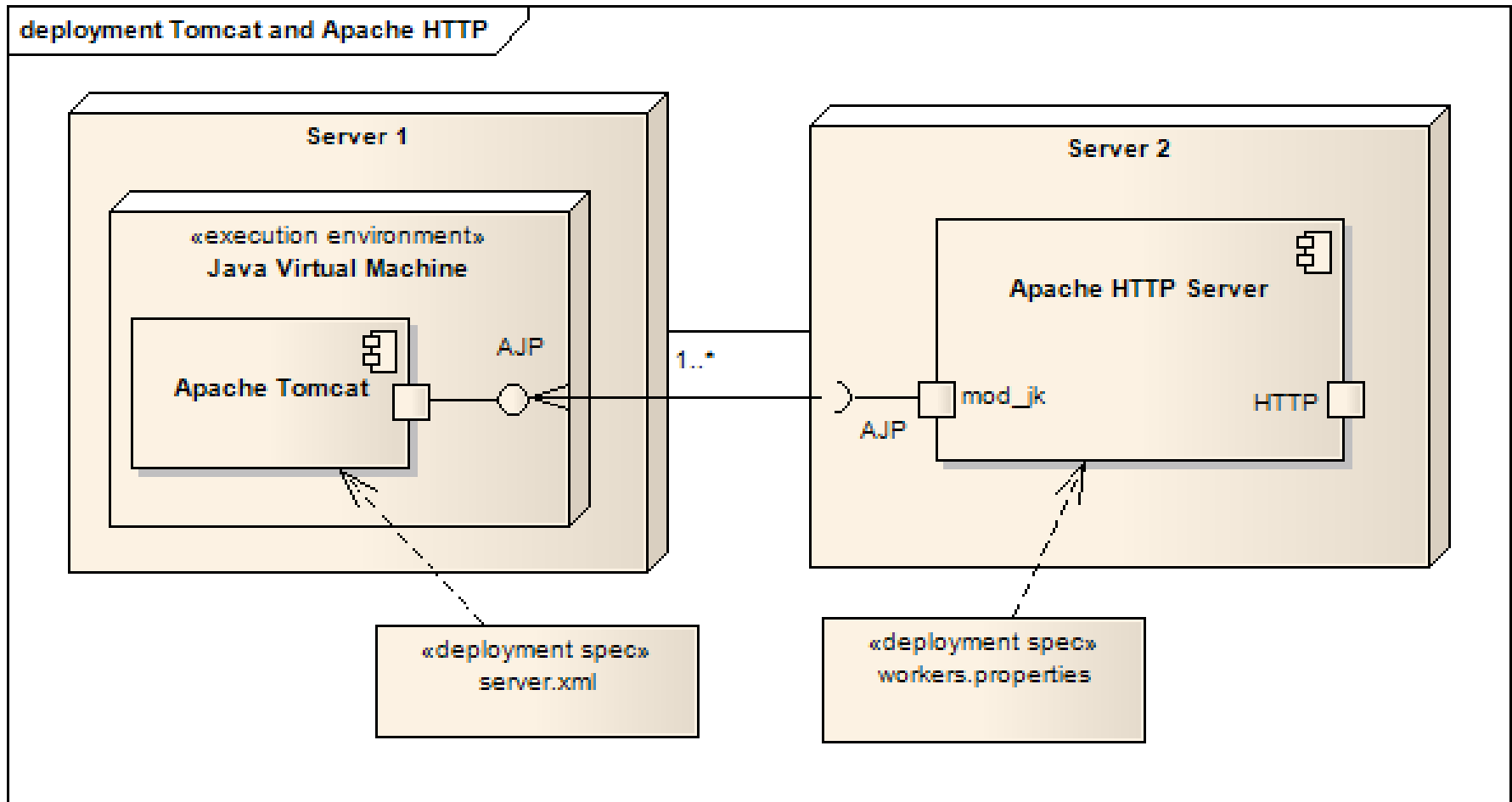
Die **Verteilungssicht** beschreibt, welche physikalischen Informationseinheiten (Jar-Files, DLLs, ...) im Rahmen des Entwicklungsprozesses erstellt bzw. benötigt werden, welche Komponenten sie manifestieren, und wie sie für den Betrieb zu verteilen sind.

Verteilungssicht und UML

- Die UML bietet eigene Modellelemente und ein Diagramm für die Verteilungssicht an
 - Verteilungsdiagramm
 - Knoten, Artefakte

UML Deployment Diagram

Beispiel: Szenario: Tomcat + Apache HTTP Server



Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3** Die Aufgabe
- 4** Sichten
- 5** Lochen und abheften
- 6** Schluss und Aus(-blick)

Dokumentation == Zusammenstellen + Ordnen

Do|ku|men|ta|ti|on [...zion] [lat.] die; -, -en: 1. a) Zusammenstellung u. Ordnung von Dokumenten und Materialien jeder Art, durch die das Benutzen und Auswerten ermöglicht oder erleichtert wird ...



Eine feste Struktur kann Orientierung geben, und hilft es sich leichter zurecht zu finden.

arc42 – Vorschlag für ein Template

(Gernot Starke, Peter Hruschka)

The screenshot shows the homepage of arc42 in a Mozilla Firefox browser window. The browser title is "arc42 - Home - Mozilla Firefox". The address bar shows "http://www.arc42.de/". The website has a blue header with the text "arc42 Ressourcen für Software-Architekten" and the arc42 logo. Below the header, there are three main columns: "Home" with a list of links, "Über arc42" with two paragraphs of text, and "News" with two news items. At the bottom, there is a Creative Commons license notice and a language selector.

Home

- Das Template
- Der Prozess
- Kurse
- Downloads
- Häufige Fragen (FAQ)
- Services
- Bibliothek
- Über...
- Sitemap

Über arc42

arc42 unterstützt Software- und Systemarchitekten. Es kommt aus der Praxis und basiert auf Erfahrungen internationaler Architekturprojekte und Rückmeldungen vieler Anwender.

arc42 enthält ein praktisches und pragmatisches Template zur Entwicklung, Dokumentation und Kommunikation von Software-Architekturen.

arc42 schlägt einen Prozess zur Entwicklung und Konstruktion effektiver Software-Architekturen vor.

News

Template auf Version 5 aktualisiert!

Verbesserungsvorschläge und Fehler können Sie hier (bei BitBucket) abgeben!

Unsere Vorträge der OOP-2011

Unsere nächsten gemeinsamen öffentlichen Kurse:

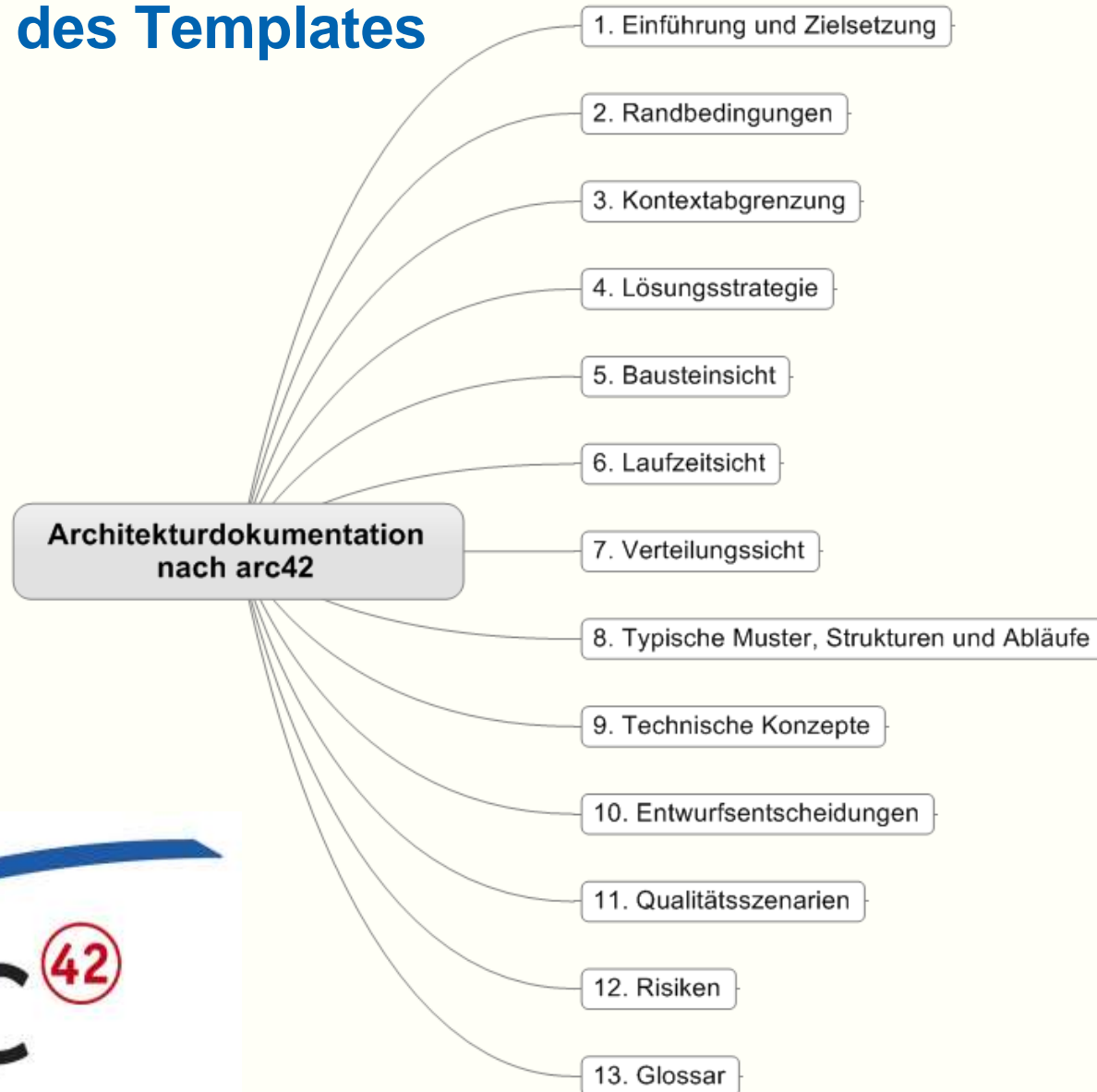
- 6.-9. Dezember 2011 in München
- 19.-22. März 2012 in München

Dieser Inhalt ist unter einer Creative Commons Lizenz lizenziert.

English 

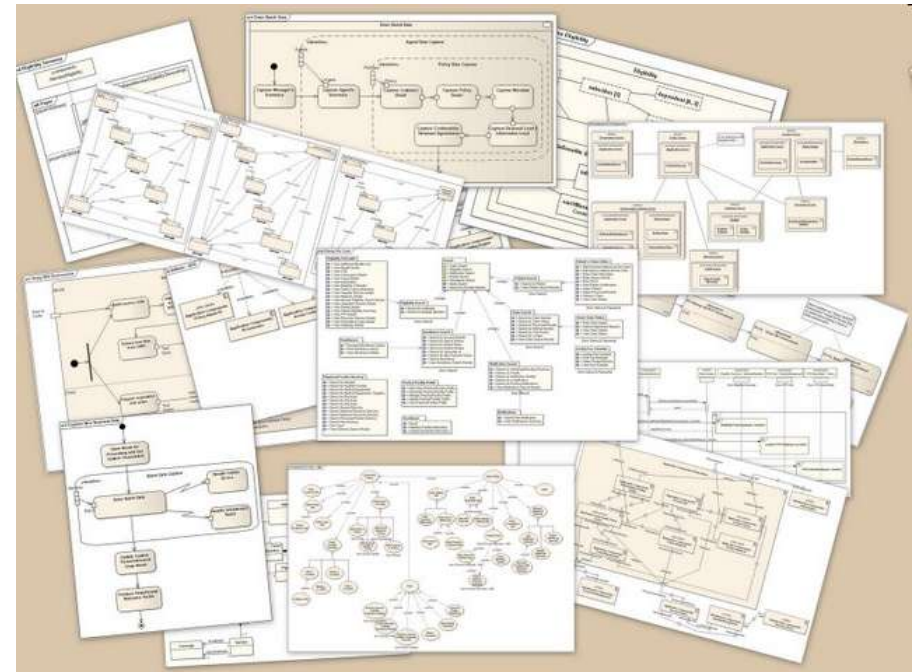
→ <http://arc42.de/>

Struktur des Templates



UML = Unified Modeling Language

- etablierte, standardisierte Notation im Bereich Software-Engineering
- Primäre Disziplinen:
 - Analyse
 - Entwurf / Architektur
- umfangreich, 14 Diagrammtypen



➔ <http://www.uml.org/>

Es muss nicht immer ein digitales Tool sein ...



Übungsergebnisse aus einem oose-Seminar zu Softwarearchitektur

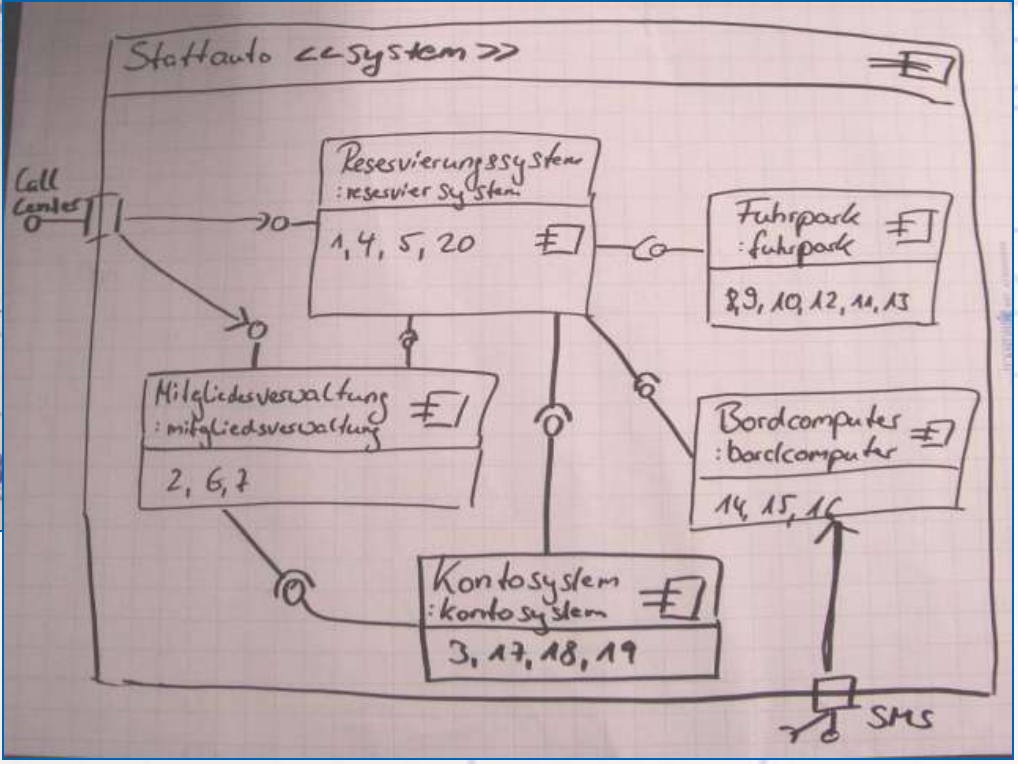
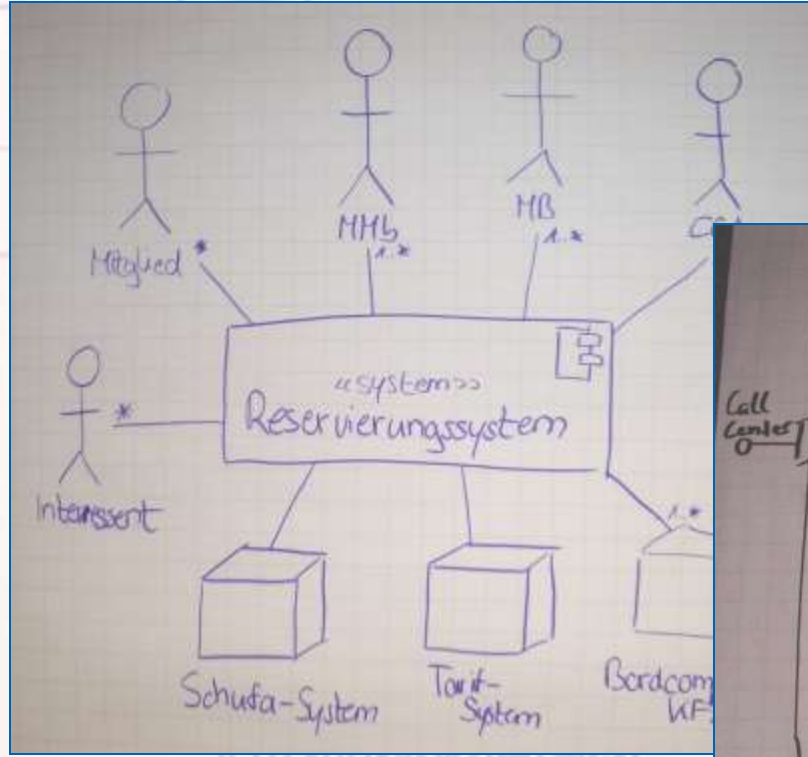
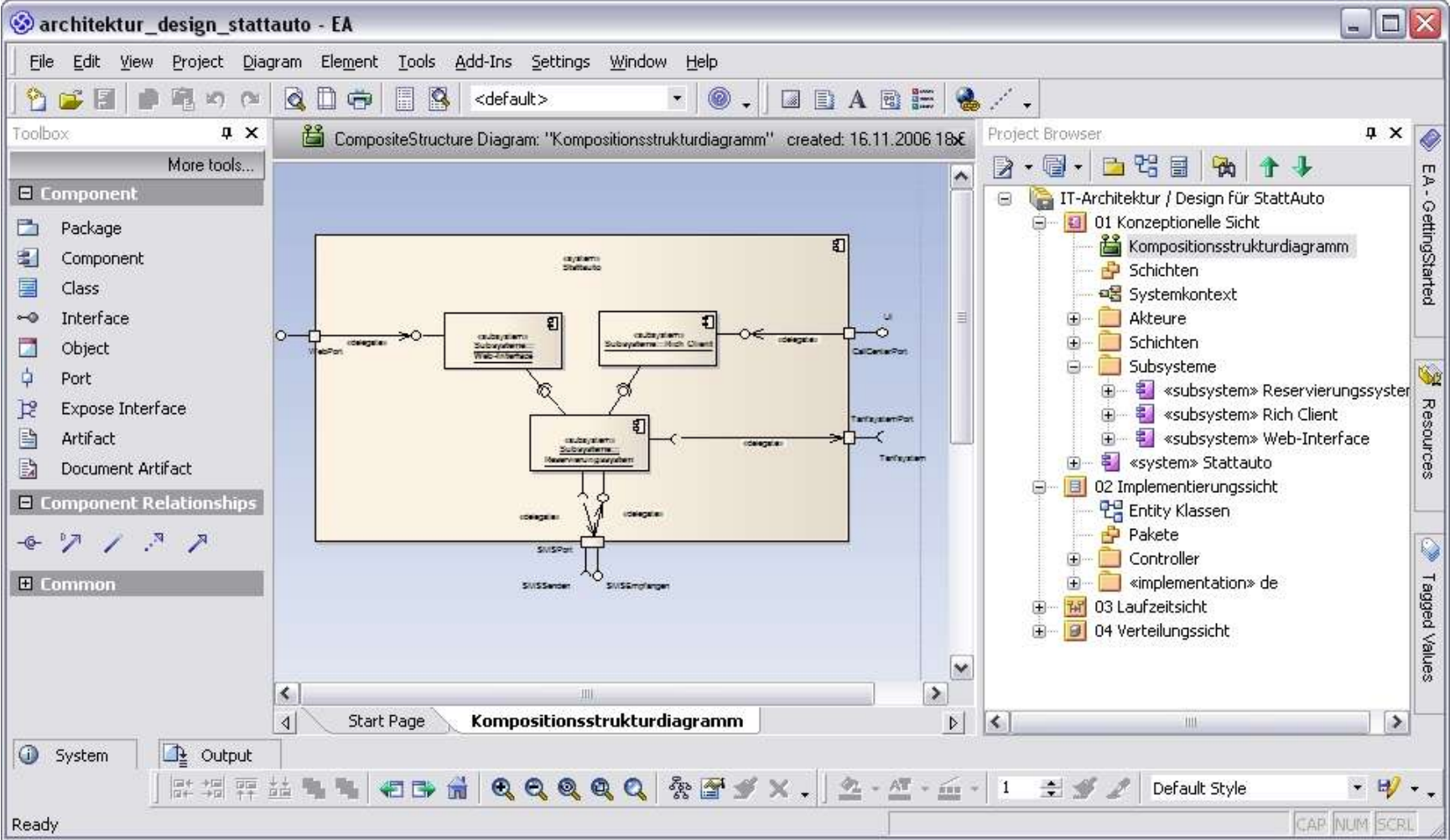


Diagramme == Sichten auf ein Modell



Und im Wiki?

„Wikis ermöglichen das gemeinschaftliche Arbeiten an Texten. Ziel eines Wikis ist es im Allgemeinen, die Erfahrung und den Wissensschatz der Autoren **kollaborativ** auszudrücken.“

wikipedia.de

- Nachvollziehbarkeit von Ergänzungen und Änderungen
 - Autor, Historie, ...
 - Benachrichtigungen
- Freies Verknüpfen von Inhalten (Links, Tags)
- Leicht zugänglich für das ganze Team (kein spezieller Client)
- Lädt zum Kommentieren ein



Generell ein tolles Medium für Entwicklungsprojekte, um untereinander zu kommunizieren.

arc42 in einem Wiki?

close close others edit more

Kontextabgrenzung

StefanZ, 6 April 2011 (created 5 April 2011)

Die Kontextsicht grenzt das System, für das Sie die Architektur entwickeln, von allen Nachbarsystemen ab. Sie legt damit die wesentlichen externen Schnittstellen fest.

```
graph TD
    WC[Web Client] ---|HTTP(S) 0..*| Tomcat["«system» Apache Tomcat"]
    Admin[Administrator] --- Tomcat
    JMX[JMX Management Client] ---|JMX| Tomcat
    Other[Anderer HTTP Server] ---|AJP| Tomcat
    Backend[Backend System] ---|JDBC, JMS, JCA, ...| Tomcat
    UserDB[User Database] ---|Realm| Tomcat
```

no tags

search

close all

perma view

new tid dler

new journal

save changes

options »

These *InterfaceOptions* for customising *TiddlyWiki* are saved in your browser

Your username for signing your edits. Write it as a *WikiWord* (eg *JoeBloggs*)

StefanZ

SaveBackups

AutoSave

RegExpSearch

CaseSensitiveSearch

EnableAnimations

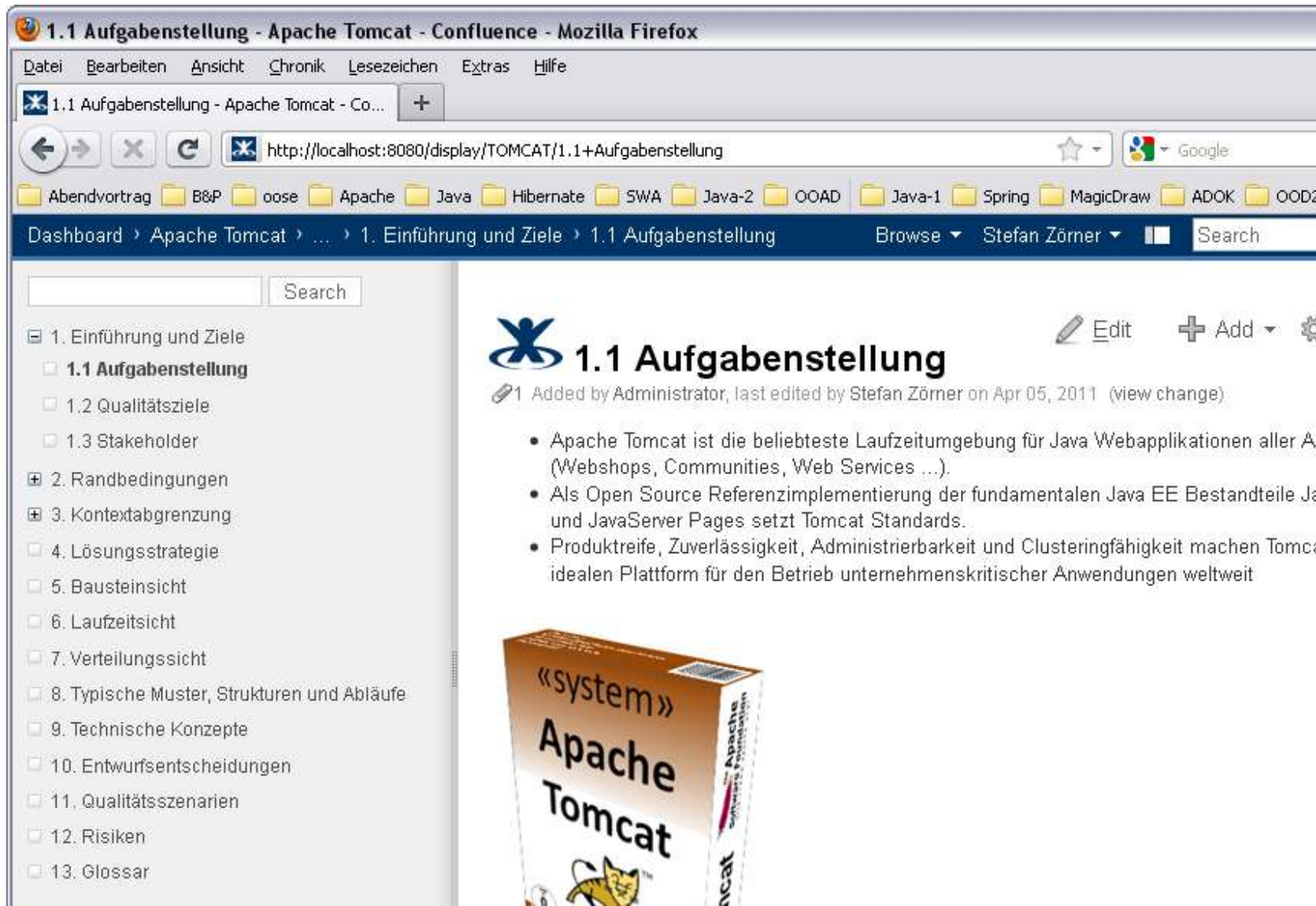
Also see *AdvancedOptions*

Timeline All Tags More

entscheidung (3)

close close others edit more

Alternatives Beispiel – Confluence



1.1 Aufgabenstellung - Apache Tomcat - Confluence - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

1.1 Aufgabenstellung - Apache Tomcat - Co... +

http://localhost:8080/display/TOMCAT/1.1+Aufgabenstellung

Abendvortrag B&P oose Apache Java Hibernate SWA Java-2 OOAD Java-1 Spring MagicDraw ADOK OOD2

Dashboard > Apache Tomcat > ... > 1. Einführung und Ziele > 1.1 Aufgabenstellung

Browse Stefan Zörner Search

Search





1. Einführung und Ziele


- 1.1 Aufgabenstellung
- 1.2 Qualitätsziele
- 1.3 Stakeholder

2. Randbedingungen

3. Kontextabgrenzung


- 4. Lösungsstrategie
- 5. Bausteinsicht
- 6. Laufzeitsicht
- 7. Verteilungssicht
- 8. Typische Muster, Strukturen und Abläufe
- 9. Technische Konzepte
- 10. Entwurfsentscheidungen
- 11. Qualitätsszenarien
- 12. Risiken
- 13. Glossar

 **1.1 Aufgabenstellung**  Edit  Add 

 1 Added by Administrator, last edited by Stefan Zörner on Apr 05, 2011 (view change)

- Apache Tomcat ist die beliebteste Laufzeitumgebung für Java Webapplikationen aller A (Webshops, Communities, Web Services ...).
- Als Open Source Referenzimplementierung der fundamentalen Java EE Bestandteile Ja und JavaServer Pages setzt Tomcat Standards.
- Produktreife, Zuverlässigkeit, Administrierbarkeit und Clusteringfähigkeit machen Tomcat idealen Plattform für den Betrieb unternehmenskritischer Anwendungen weltweit

«system»
Apache
Tomcat



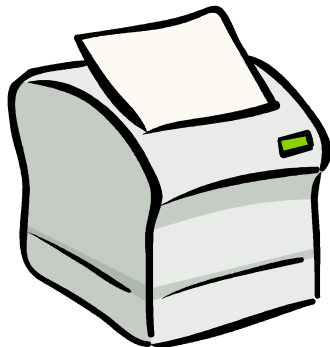
Herausforderungen

Diagramme

- Wie erstellt man Abbildungen im bzw. für das Wiki
- Wie hält man Abbildungen und Textinhalte konsistent?

Versionierung

- Wikis führen Versionen für einzelne Seiten
- Wie versioniert man die Dokumentation vollständig, z.B. für ein Release?



Drucken

- Wie gibt man die Dokumentation aus dem Wiki als Dokument (z.B. PDF) heraus?
- Wie befüllt man eine vorgegebene Struktur?

Entscheidungsfaktoren



Agenda

- 1** Motivation – Montag Morgen
- 2** Softwarearchitektur
- 3** Die Aufgabe
- 4** Sichten
- 5** Lochen und abheften
- 6** Schluss und Aus(-blick)

These.

Wenn Ihr beim Entwurf Ihrer Softwarearchitektur methodisch vorgeht, fallen zentrale Ergebnisse für die Dokumentation automatisch ab.



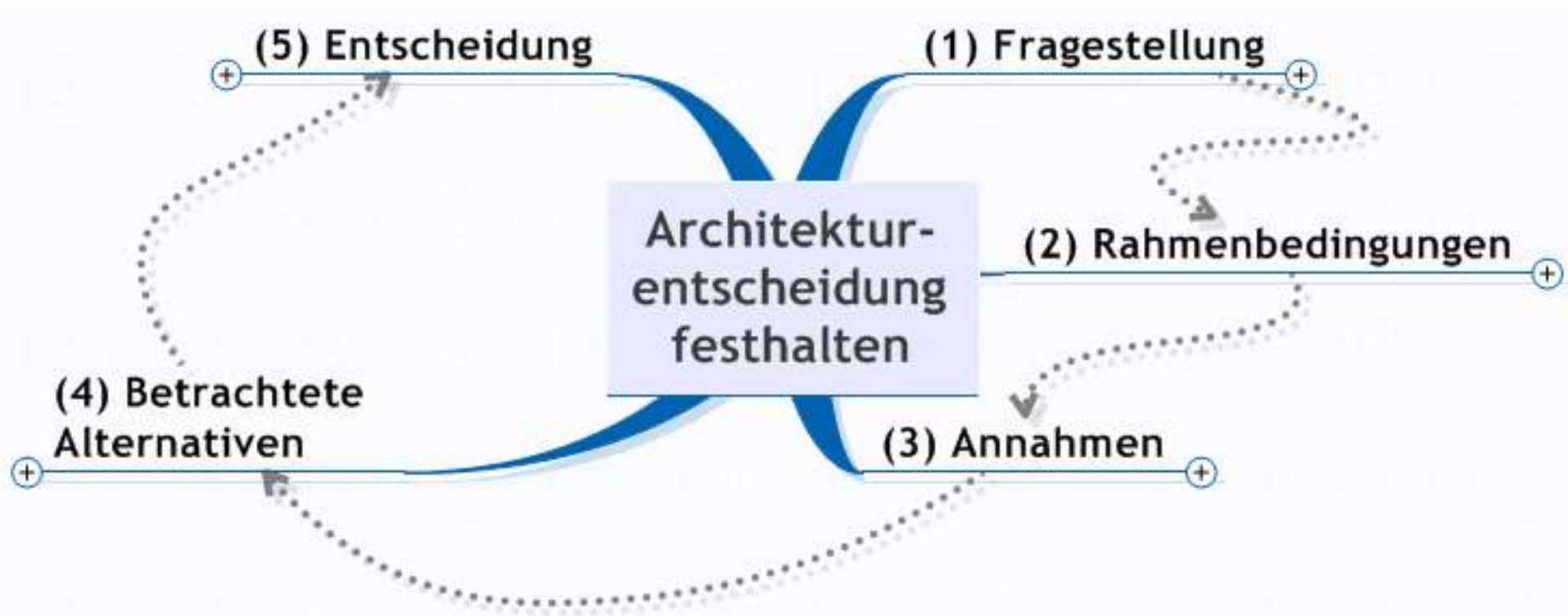
Architekturentscheidungen ...



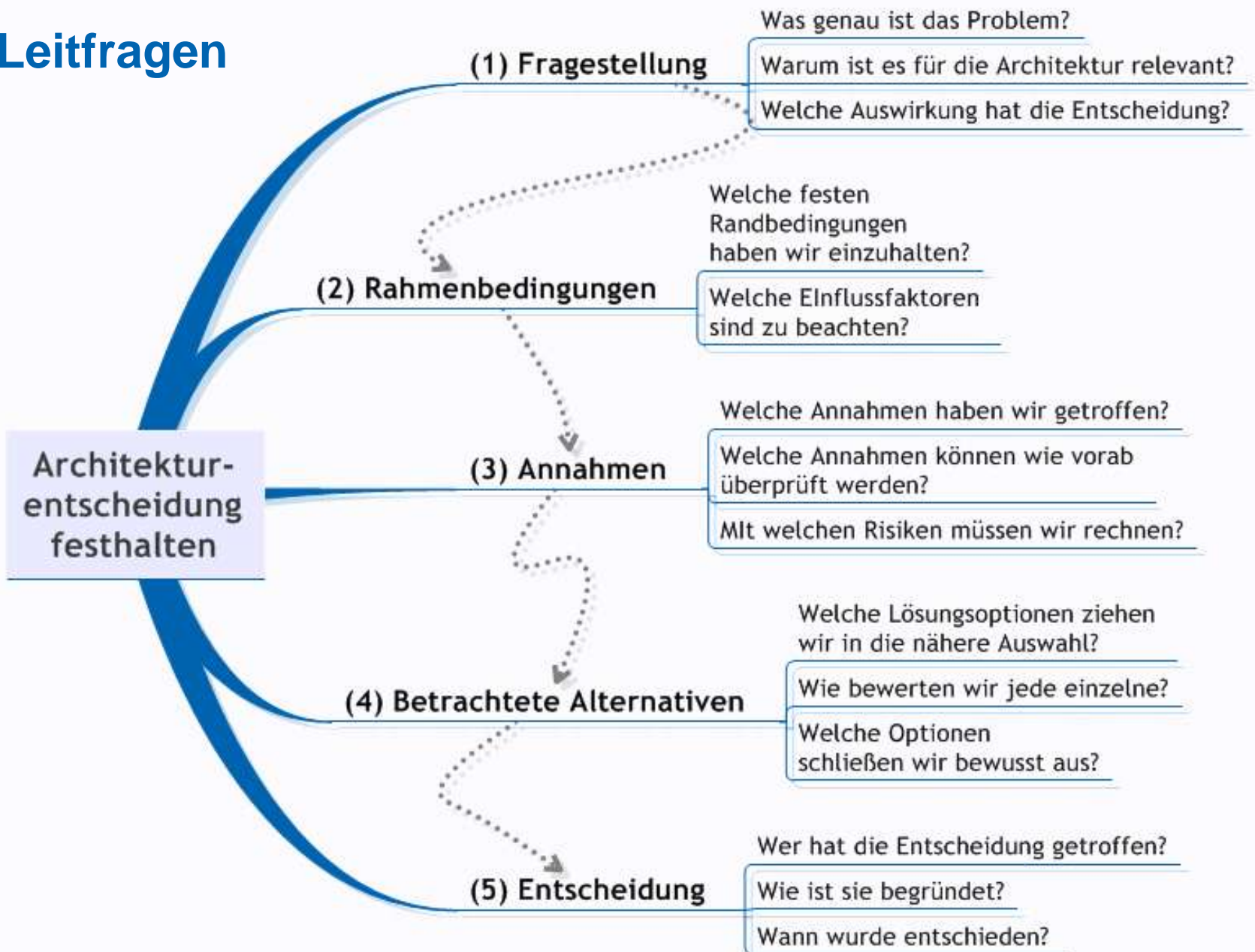
“Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.” (Eoin Woods)

- Architekturentscheidungen zählen zu den Dingen, die sich im nachhinein kaum bis gar nicht dokumentieren lassen.

Entscheidungen treffen und festhalten. Ein Werkzeug



Leitfragen



Auch im Web

Architekturen dokumentieren: Kombiniere - der Systemkontext | IT-Republik - JAXenter - News - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://it-republik.de/jaxenter/news/Architekturen-dokumentieren-Kombiniere---der-System

it republik » JAXenter

Magazine Konferenzen Bücher Akademie Fo

HOME

Jaxenter
Portal für Java™, Enterprise Architekturen, SOA

it republik Java Magazin Eclipse Magazin JAX Business Technology Forum

JAXENTER ▾

bookmarken | drucken | empfehlen | kommentieren

Montag, 27. Oktober 2008 | Kolumne

Architekturen dokumentieren: Kombiniere - der Systemkontext

Waren Sie schon einmal im Krimitheater? Samstagabend, 20:00 Uhr: Auf der Bühne...

Performanceumfrage
Mitmachen und gewi

NEWSLETTER
Täglich aktuelle News
E-Mail-Adress

JETZT IM HANDEL
Partner
ecli

→ <http://it-republik.de/jaxenter/>

Softwarearchitektur – Seminare bei oose.

Als öffentliche Veranstaltungen in Hamburg, oder vor Ort bei Ihnen, z.B.

- [SWA] Praktische Softwarearchitektur - Lebendiger Entwurf mit Methode
- [SWA+Zert] inkl. Aufbaukurs iSAQB-CPSA und Zertifizierung
- [OOD2] Fortgeschrittenes Design – Entwurfsprinzipien und –muster
- [AWERT] Software-Architekturbewertung
- [ADOK] Softwarearchitekturen festhalten und kommunizieren

➔ <http://www.oose.de>



Vielen Dank!

Ich freue mich auf Eure Fragen ...



Stefan Zörner :: Stefan.Zoerner@oose.de