

# Zwischenbericht

KONFERENZSYSTEM

PATRICK ROBINSON, BASTIAN MAGER

## Änderungsübersicht

Version	Datum	Bearbeiter	Beschreibung
1	12.11.2014	Patrick Robinson Bastian Mager	Initialer Entwurf
2	13.11.2014	Patrick Robinson	Format überarbeitet Komponenten detaillierter beschrieben
3	14.11.2014	Bastian Mager	Use-Case Diagramm in Kapitel 3 eingefügt
4	12.12.2014	Bastian Mager	Änderungen für die Anwendungsfälle „Einrichtung einer Konferenz“ und „Registrierung bei Konferenz“ (Übung 8)
5	20.03.2015	Bastian Mager	Entwurfsentscheidung und neue Diagramme
6	21.03.2015	Bastian Mager	Use-Case Diagramm überarbeitet
7	21.03.2015	Patrick Robinson	Komponentenentwurf & -beschreibung überarbeitet Entwurfsentscheidung erweitert
8	22.02.2015	Bastian Mager	kleine Anpassungen

## Inhalt

Änderungsübersicht .....	1
1. Einführung und Ziele .....	3
1.1. Aufgabenstellung.....	3
1.2. Qualitätsziele .....	3
1.3. Stakeholder.....	3
2. Randbedingungen .....	4
2.1. Technische Randbedingungen .....	4
2.2. Organisatorische Randbedingungen .....	4
2.3. Konventionen .....	4
3. Kontextabgrenzung .....	5
3.1. Fachliche Abgrenzung.....	5
3.2. Technischer Kontext .....	5
4. Lösungsstrategie.....	6
5. Bausteinsicht .....	7
5.1. Ebene 1 .....	7
5.1.1. Komponentenschichten .....	7
5.1.2. Komponentenschichten (detailliert) .....	7
5.1.3. Komponenten mit Schnittstellen .....	8
5.2. Ebene 2 .....	12
5.2.1. Entitäten .....	12
6. Verteilungssicht.....	13
6.1. Verteilungsdiagramm .....	13
7. Konzepte.....	14
7.1. Ablaufsteuerung (Page-Flow) .....	14
7.2. Ausnahme- und Fehlerbehandlung .....	14
8. Entwurfsentscheidungen .....	15
8.1. Eigene Implementierung für Authentifizierung und Autorisierung .....	15
8.2. CQRS & Anemic Domain Model.....	15
9. Risiken .....	16

# 1. Einführung und Ziele

## 1.1. Aufgabenstellung

Ziel ist es eine Konferenzverwaltungssoftware mit Fokus auf Erstellen und Begutachten von Publikationen zu entwickeln.

## 1.2. Qualitätsziele

Prio	Ziel	Beschreibung
1.	Sicherheit	Benutzerlogin, Rollenbasiertes Konzept für Rechte

## 1.3. Stakeholder

Rolle	Beschreibung
Veranstalter	Will erfolgreiche Konferenz organisieren
Autor	Wissenschaftler, der Erkenntnisse/Forschungsergebnisse weitergeben möchte Veröffentlicht Publikation(-en) Will gutes Gutachten bekommen
Gutachter	Liest Publikationen Erstellt Gutachten
Teilnehmer	Liest Publikationen Will neue Erkenntnisse gewinnen
Entwickler	

## 2. Randbedingungen

### 2.1. Technische Randbedingungen

Hardwarevorgaben	
Softwarevorgaben	Glassfish
	JPA basierte Persistenz
	JSF basiert Weboberfläche
Vorgaben des Systembetriebs	
Programmiervorgaben	JavaEE (EJB, CDI, JSF)

### 2.2. Organisatorische Randbedingungen

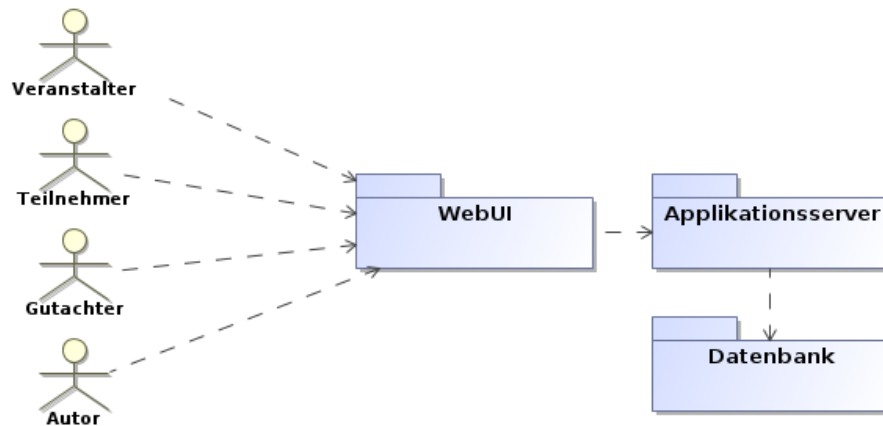
Organisation und Struktur	
Ressourcen	
Organisatorische Standards	
Juristische Faktoren	

### 2.3. Konventionen

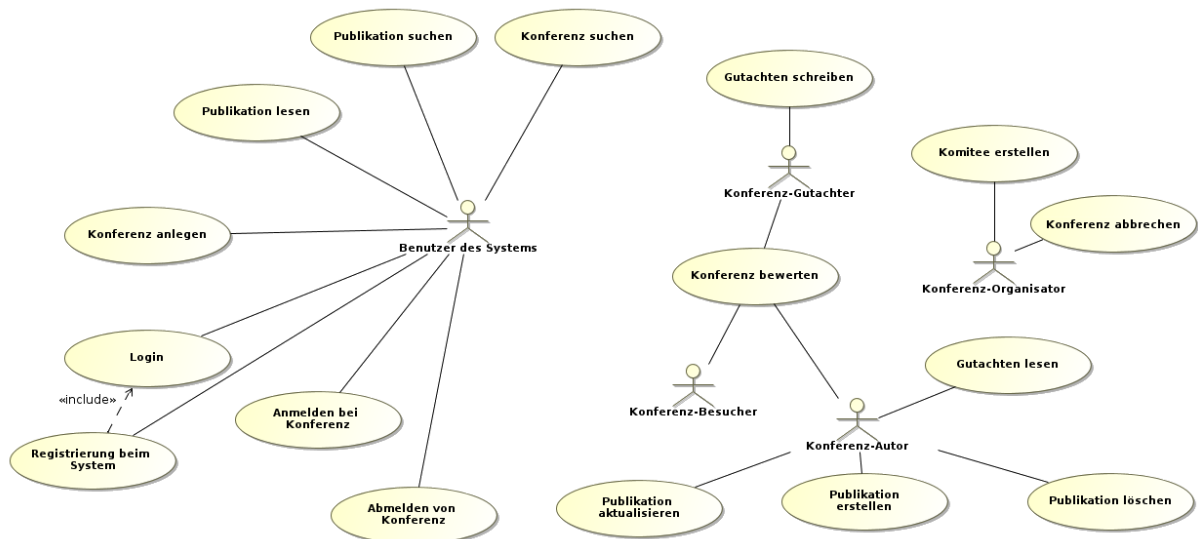
Programmierrichtlinien	
Dokumentationsrichtlinien	Dokumentation mit arc42
	Modellierung nach 4-Sichten
Richtlinien für Versions- und Konfigurationsmanagement	
Namenskonventionen	

## 3. Kontextabgrenzung

### 3.1. Fachliche Abgrenzung



Die folgende Abbildung zeigt die fachlichen Anforderungen an das System in Zusammenhang mit den Akteuren in Form eines Use-Case-Diagramms:



### 3.2. Technischer Kontext

Nachbar	Schnittstelle	Technisches Protokoll	Datenformat
PostgreSQL	JPA	EclipseLink, JDBC	Autogeneriert von JPA

## 4. Lösungsstrategie

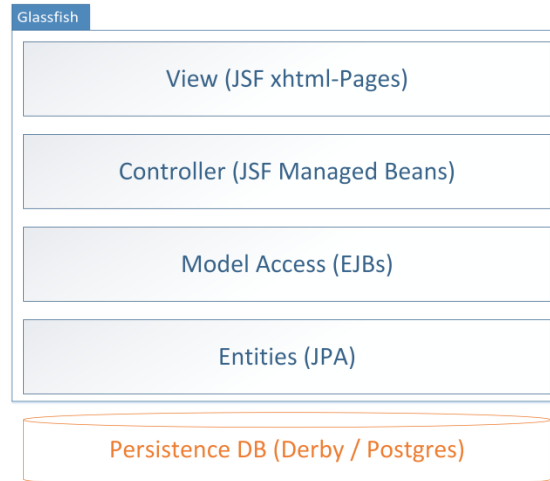
Das System ist auf oberster Ebene in zwei Teile geteilt. Ein GUI-unabhängiges Model, das aus reinen Daten und Datenverarbeitung besteht, und GUI-spezifischen Code, der die Daten darstellt und Aktionen der Nutzer überprüft und an das Model weiterleitet.

Programmierung erfolgt in Netbeans, mit dessen Hilfe auch die Entity-Klassen generieren werden. Initial werden mit SQL die Relationen definiert und anschließend die Entity-Klassen von Netbeans generiert. Sukzessiv werden dann einige Anpassungen vorgenommen. Weiterhin wird JPA für JDBC und ORM verwendet.

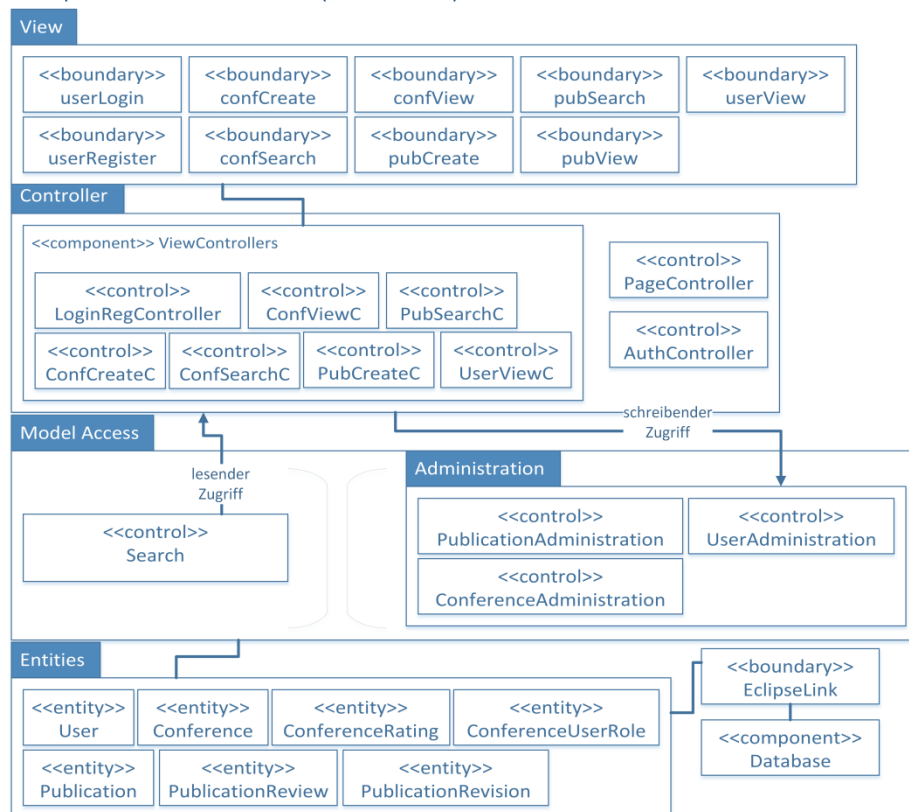
## 5. Bausteinsicht

### 5.1. Ebene 1

#### 5.1.1. Komponentenschichten

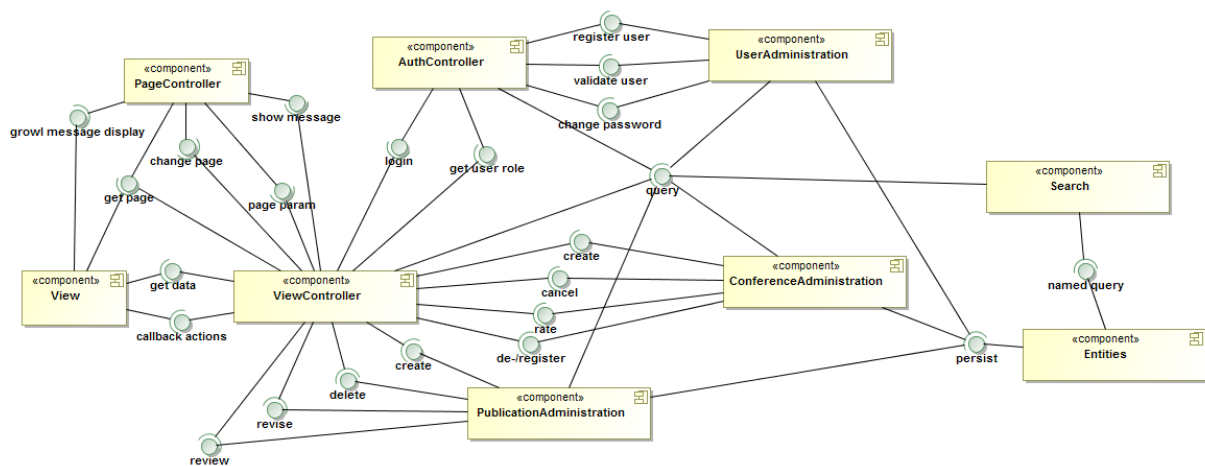


#### 5.1.2. Komponentenschichten (detailliert)





### 5.1.3. Komponenten mit Schnittstellen



#### 5.1.3.1. View

<b>Zweck</b>	Schnittstelle für die Benutzerinteraktion	
<b>Schnittstellen</b>	Benötigt	Seitenspezifischer ViewController - Stellt aufbereitete Daten bereit - Stellt Rückrufmethoden bereit get page
	Bietet	growl message display
<b>Ablageort</b>	*.xhtml	
<b>Erfüllte Anforderungen</b>	JSF basierte Weboberfläche	

#### 5.1.3.2. View Controller

<b>Zweck</b>	Controller für jeweilige Seite	
<b>Schnittstellen</b>	Benötigt	create publication delete publication revise publication review publication login get user role create conference cancel conference rate conference register to conference deregister from conference show message page parameters change page get page
	Bietet	get data callback actions
<b>Ablageort</b>	ConferenceCreateController.java	
	ConferenceSearchController.java	
	ConferenceViewController.java	
	PublicationCreateController.java	

	PublicationSearchController.java
	PublicationViewController.java
	UserLoginRegController.java
	UserViewController.java
<b>Erfüllte Anforderungen</b>	

#### 5.1.3.3. Search

<b>Zweck</b>	Auf Teile des Datenbestands lesend zugreifen	
<b>Schnittstellen</b>	Bietet	get conference by id
		get all conferences
		search conferences organized by
		search conferences by name starting with
		search conference roles for user
		search users for conference
		get average rating of conference
		search for users
		search user by id
		search organizer for conference
		get average rating of organizer
		search for publication
		get all publications
		search publications for user
		search publications by title starting with
		search reviews for user
		search reviews for publication
	Benötigt	Named Queries
<b>Ablageort</b>	Search.java, SearchLocal.java	
<b>Erfüllte Anforderungen</b>		

#### 5.1.3.4. Page Controller

<b>Zweck</b>	Globale Interaktion mit Webbrowser	
<b>Schnittstellen</b>	Bietet	Show message
		Page parameters
		Change page
		Get page
	Benötigt	Growl message display
<b>Ablageort</b>	PageController.java	
<b>Erfüllte Anforderungen</b>		

#### 5.1.3.5. Authentication Controller

<b>Zweck</b>	Nutzeridentifikation und Berechtigungsverwaltung	
<b>Schnittstellen</b>	Bietet	Get user role
		Login
	Benötigt	Register user
		Validate user

	Change password
<b>Ablageort</b>	AuthenticationController.java, AuthenticatedViewController.java
<b>Erfüllte Anforderungen</b>	

#### 5.1.3.6. User Administration

<b>Zweck</b>	Änderungen an Benutzern durchführen	
<b>Schnittstellen</b>	Bietet	Register user
		Validate user
		Change password
	Benötigt	Persist
		Query
<b>Ablageort</b>	UserAdministration.java, UserAdministrationLocal.java	
<b>Erfüllte Anforderungen</b>		

#### 5.1.3.7. Publication Administration

<b>Zweck</b>	Änderungen an Publikationen durchführen	
<b>Schnittstellen</b>	Bietet	Create publication
		Delete publication
		Revise publication
		Review publication
	Benötigt	Persist
		Query
<b>Ablageort</b>	PublicationAdministration.java, PublicationAdministrationLocal.java	
<b>Erfüllte Anforderungen</b>		

#### 5.1.3.8. Conference Administration

<b>Zweck</b>	Änderungen an Konferenzen durchführen	
<b>Schnittstellen</b>	Bietet	Create conference
		Cancel conference
		Rate conference
		Register to conference
		Deregister from conference
	Benötigt	Persist
		Query
<b>Ablageort</b>	ConferenceAdministration.java, ConferenceAdministrationLocal.java	
<b>Erfüllte Anforderungen</b>		

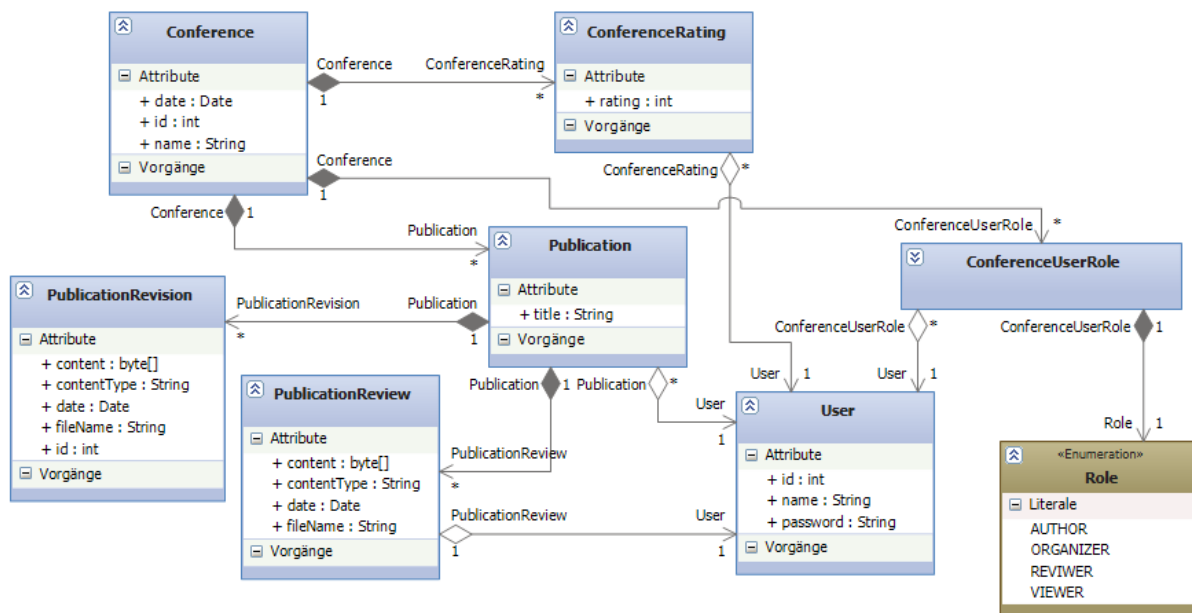
#### 5.1.3.9. Entities

<b>Zweck</b>	Persistente Speicherung der Daten	
<b>Schnittstellen</b>	Bietet	Persist
		Named queries
<b>Ablageort</b>	Conference.java	

	ConferenceRating.java
	ConferenceRatingPK.java
	ConferenceUserRole.java
	ConferenceUserRolePK.java
	Publication.java
	PublicationPK.java
	PublicationReview.java
	PublicationReviewPK.java
	PublicationRevision.java
	PublicationRevisionPK.java
	User.java
<b>Erfüllte Anforderungen</b>	JPA basierte Persistenz

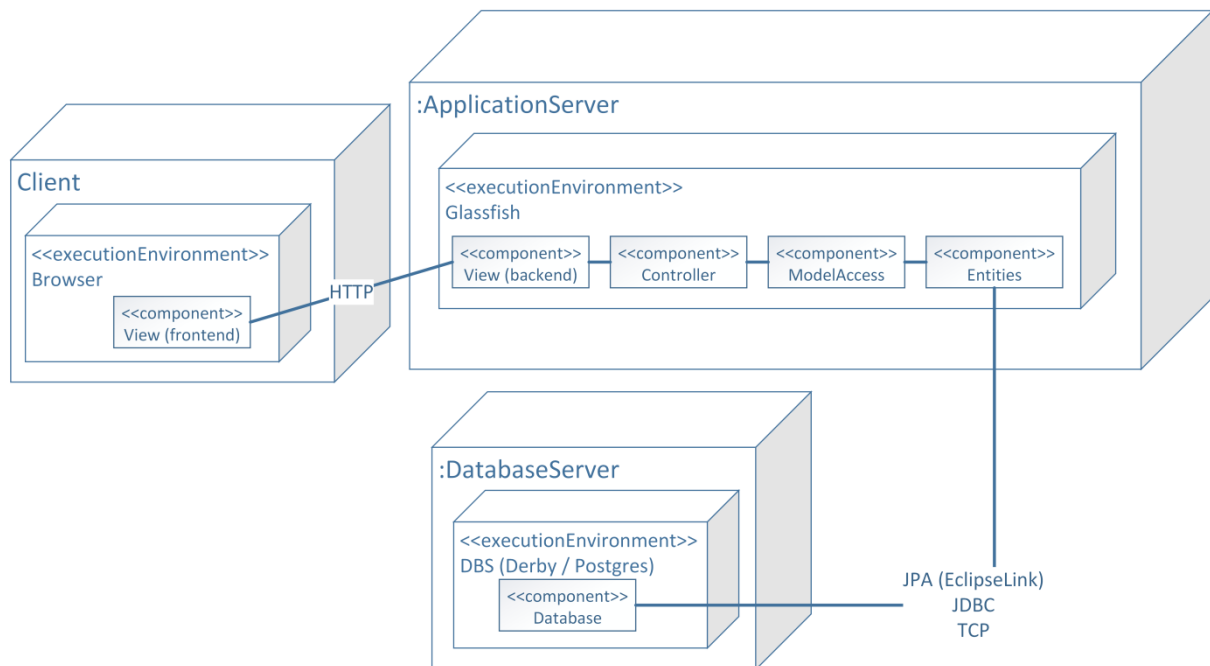
## 5.2. Ebene 2

### 5.2.1. Entitäten



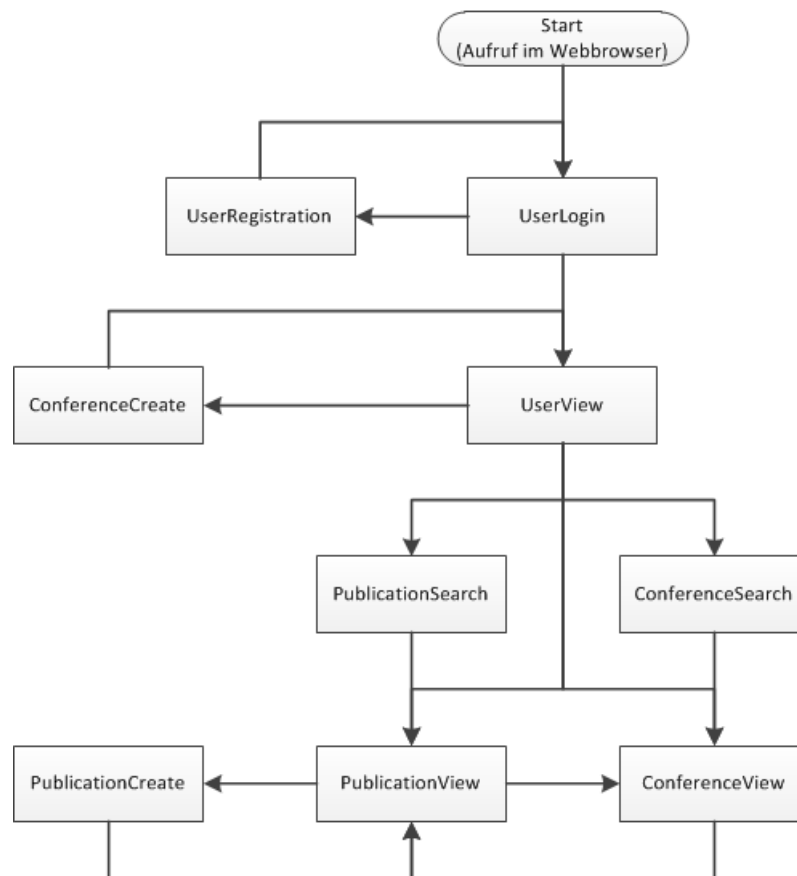
## 6. Verteilungssicht

### 6.1. Verteilungsdiagramm



## 7. Konzepte

### 7.1. Ablaufsteuerung (Page-Flow)



### 7.2. Ausnahme- und Fehlerbehandlung

Fehler werden dem Benutzer aufbereitet als Growl-Notifications<sup>1</sup> angezeigt, wenn dies angebracht ist, z.B. da der Nutzer den Fehler beheben kann (z.B. falsches Kennwort). Diese besteht aus Titel, Erklärung und einem Schweregrad, der durch Symbole repräsentiert wird.

Da es keine Administratoren gibt findet keine weitere Protokollierung statt.

<sup>1</sup> Vgl. <http://www.primefaces.org/showcase/ui/message/growl.xhtml>

## 8. Entwurfsentscheidungen

### 8.1. Eigene Implementierung für Authentifizierung und Autorisierung

Nach Einschätzung ist das Konzept von Java EE bzw. Glassfish zur Umsetzung der Authentifizierung und Autorisierung ungeeignet.

In dem System ist jeder Benutzer zunächst gleich berechtigt (also kein genereller Unterschied zwischen Rollen Administrator und Nutzer). Die Rollen der angemeldeten Benutzer sind jedoch für jede Konferenz anders (bzw. wechseln ständig je nach View). Nicht eine Rolle pro Benutzer, sondern eine Rolle pro Verbindung zwischen Konferenz und Benutzer.

Es werden also nicht ganze Bereiche für bestimmte Rollen abgeschirmt. Die Rollenverwaltung von Java EE gibt jedem Nutzer eine Rolle und man kann dann Methoden nur für spezifische Rollen zugreifbar machen.

### 8.2. CQRS & Anemic Domain Model

Das Model ist in zwei Teile geteilt. Die Daten (in Entities) sind von den verarbeitenden Methoden (in EJBs) getrennt.

Die verarbeitenden Methoden des Models sind wiederum strikt in lesenden und schreibenden Zugriff getrennt.



## 9. Risiken

Priorität	Risiko	Erläuterung
1	Zeitmanagement	Einhaltung von Terminen