Bericht

Konferenzsystem

Patrick Robinson, Bastian Mager

2015

# 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 | 22.03.2015 | Bastian Mager | kleine Anpassungen |
| 9 | 22.03.2015 | Patrick Robinson | Diagramme durch neuste Versionen ersetzt  Titel angepasst |

Inhalt

[Änderungsübersicht 1](#_Toc414793291)

[1. Einführung und Ziele 3](#_Toc414793292)

[1.1. Aufgabenstellung 3](#_Toc414793293)

[1.2. Qualitätsziele 3](#_Toc414793294)

[1.3. Stakeholder 3](#_Toc414793295)

[2. Randbedingungen 4](#_Toc414793296)

[2.1. Technische Randbedingungen 4](#_Toc414793297)

[2.2. Organisatorische Randbedingungen 4](#_Toc414793298)

[2.3. Konventionen 4](#_Toc414793299)

[3. Kontextabgrenzung 5](#_Toc414793300)

[3.1. Fachliche Abgrenzung 5](#_Toc414793301)

[3.2. Technischer Kontext 5](#_Toc414793302)

[4. Lösungsstrategie 6](#_Toc414793303)

[5. Bausteinsicht 7](#_Toc414793304)

[5.1. Ebene 1 7](#_Toc414793305)

[5.1.1. Komponentenschichten 7](#_Toc414793306)

[5.1.2. Komponentenschichten (detailliert) 7](#_Toc414793307)

[5.1.3. Komponenten mit Schnittstellen 8](#_Toc414793308)

[5.2. Ebene 2 12](#_Toc414793309)

[5.2.1. Entitäten 12](#_Toc414793310)

[6. Verteilungssicht 13](#_Toc414793311)

[6.1. Verteilungsdiagramm 13](#_Toc414793312)

[7. Konzepte 14](#_Toc414793313)

[7.1. Ablaufsteuerung (Page-Flow) 14](#_Toc414793314)

[7.2. Ausnahme- und Fehlerbehandlung 14](#_Toc414793315)

[8. Entwurfsentscheidungen 15](#_Toc414793316)

[8.1. Eigene Implementierung für Authentifizierung und Autorisierung 15](#_Toc414793317)

[8.2. CQRS & Anemic Domain Model 15](#_Toc414793318)

[9. Risiken 16](#_Toc414793319)

# Einführung und Ziele

## Aufgabenstellung

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

## Qualitätsziele

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

## 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 |  |

# Randbedingungen

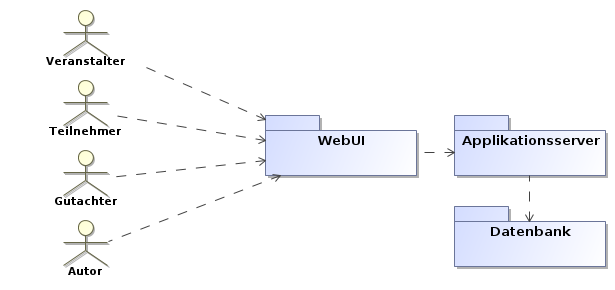
|  |  |
| --- | --- |
| 2.1. Technische Randbedingungen | |
| Hardwarevorgaben |  |
| Softwarevorgaben | Glassfish |
| JPA basierte Persistenz |
| JSF basiert Weboberfläche |
| Vorgaben des Systembetriebs |  |
| Programmiervorgaben | JavaEE (EJB, CDI, JSF) |

|  |  |
| --- | --- |
| 2.2. Organisatorische Randbedingungen | |
| Organisation und Struktur |  |
| Ressourcen |  |
| Organisatorische Standards |  |
| Juristische Faktoren |  |

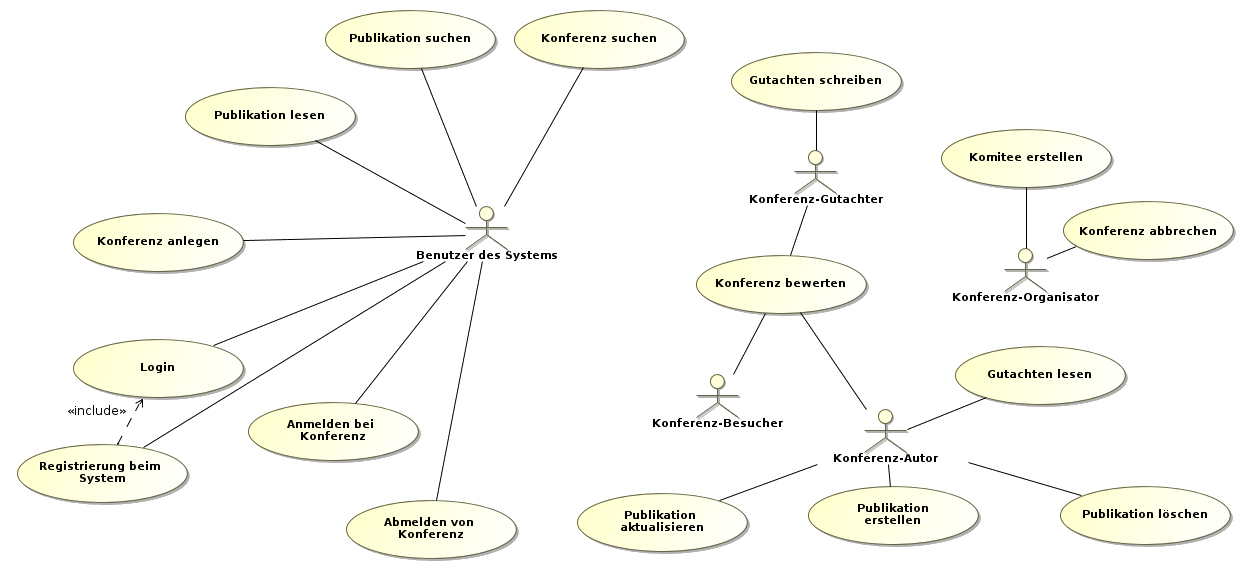
|  |  |
| --- | --- |
| 2.3. Konventionen | |
| Programmierrichtlinien |  |
| Dokumentationsrichtlinien | Dokumentation mit arc42 |
| Modellierung nach 4-Sichten |
| Richtlinien für Versions- und Konfigurationsmanagement |  |
| Namenskonventionen |  |

# Kontextabgrenzung

## Fachliche Abgrenzung



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



## Technischer Kontext

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

# Lösungsstrategie

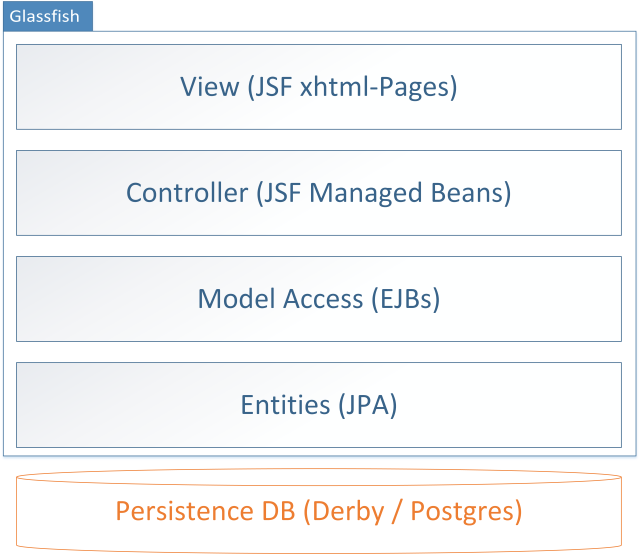
Das System ist auf oberster Ebene in zwei Teile geteilt. Ein GUI-unabhängiges Model, das aus reinen Daten und Datenverarbeitung besteht, und GUI-spezifischen Code, der die Daten darstellt und Aktionen der Nutzer überprüft und an das Model weiterleitet.

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.

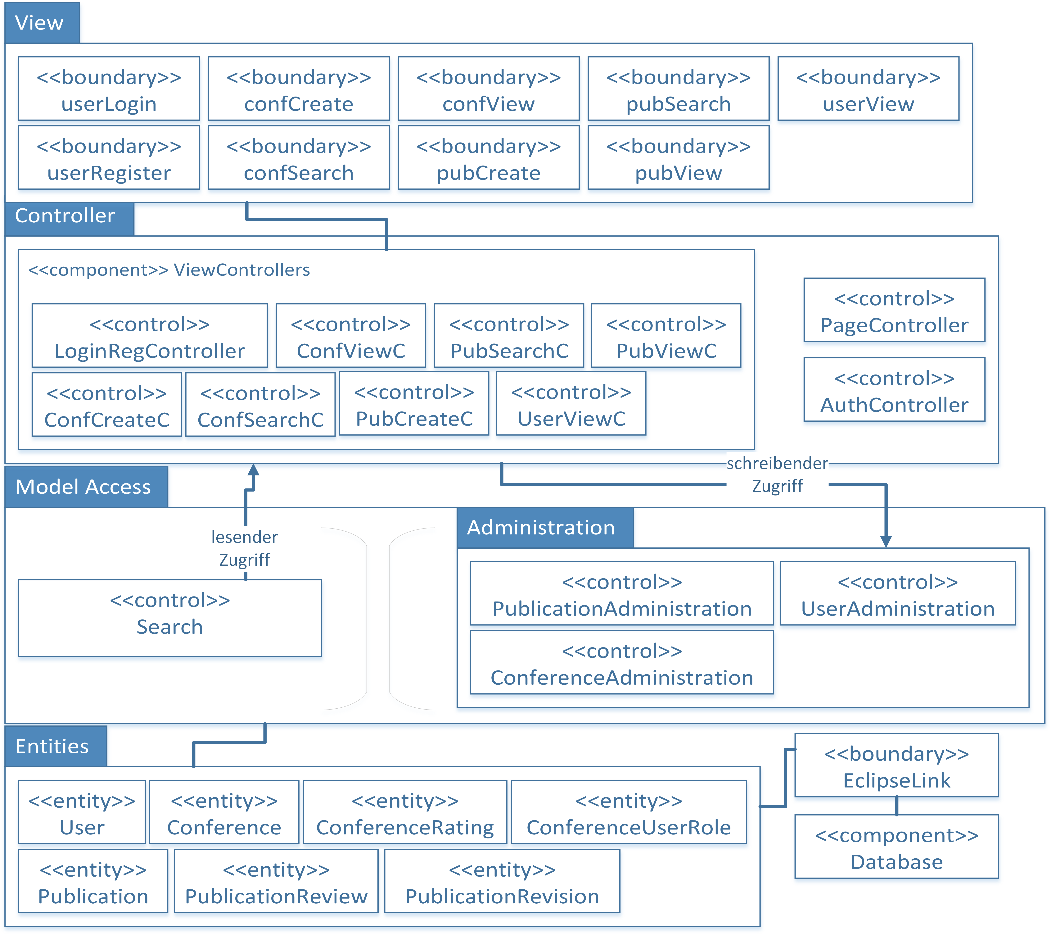
# Bausteinsicht

## Ebene 1

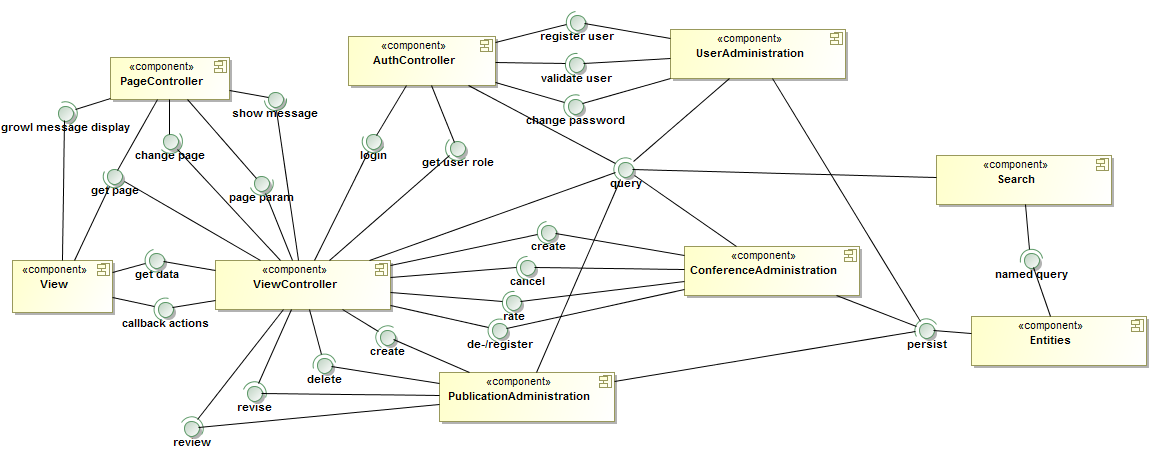
### Komponentenschichten



### Komponentenschichten (detailliert)



### Komponenten mit Schnittstellen



|  |  |  |
| --- | --- | --- |
| 5.1.3.1. View | | |
| Zweck | Schnittstelle für die Benutzerinteraktion | |
| Schnittstellen | Benötigt | Seitenspezifischer ViewController   * Stellt aufbereitete Daten bereit * Stellt Rückrufmethoden bereit |
| get page |
| Bietet | growl message display |
| Ablageort | \*.xhtml | |
| Erfüllte Anforderungen | JSF basierte Weboberfläche | |
| 5.1.3.2. View Controller | | |
| Zweck | Controller für jeweilige Seite | |
| Schnittstellen | 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 |
| Ablageort | ConferenceCreateController.java | |
| ConferenceSearchController.java | |
| ConferenceViewController.java | |
| PublicationCreateController.java | |
| PublicationSearchController.java | |
| PublicationViewController.java | |
| UserLoginRegController.java | |
| UserViewController.java | |
| Erfüllte Anforderungen |  | |

|  |  |  |
| --- | --- | --- |
| 5.1.3.3. Search | | |
| Zweck | Auf Teile des Datenbestands lesend zugreifen | |
| Schnittstellen | 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 |
| Ablageort | Search.java, SearchLocal.java | |
| Erfüllte Anforderungen |  | |
| 5.1.3.4. Page Controller | | |
| Zweck | Globale Interaktion mit Webbrowser | |
| Schnittstellen | Bietet | Show message |
| Page parameters |
| Change page |
| Get page |
| Benötigt | Growl message display |
| Ablageort | PageController.java | |
| Erfüllte Anforderungen |  | |

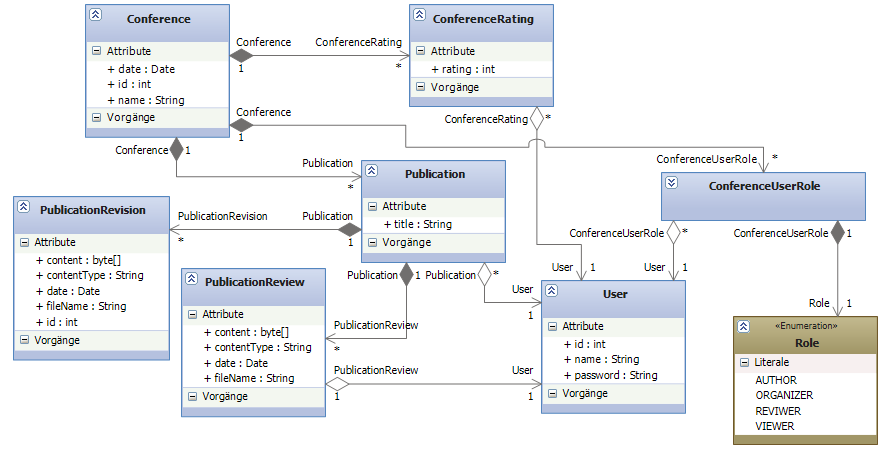
|  |  |  |
| --- | --- | --- |
| 5.1.3.5. Authentication Controller | | |
| Zweck | Nutzeridentifikation und Berechtigungsverwaltung | |
| Schnittstellen | Bietet | Get user role |
| Login |
| Benötigt | Register user |
| Validate user |
| Change password |
| Ablageort | AuthenticationController.java, AuthenticatedViewController.java | |
| Erfüllte Anforderungen |  | |

|  |  |  |
| --- | --- | --- |
| 5.1.3.6. User Administration | | |
| Zweck | Änderungen an Benutzern durchführen | |
| Schnittstellen | Bietet | Register user |
| Validate user |
| Change password |
| Benötigt | Persist |
| Query |
| Ablageort | UserAdministration.java, UserAdministrationLocal.java | |
| Erfüllte Anforderungen |  | |
| 5.1.3.7. Publication Administration | | |
| Zweck | Änderungen an Publikationen durchführen | |
| Schnittstellen | Bietet | Create publication |
| Delete publication |
| Revise publication |
| Review publication |
| Benötigt | Persist |
| Query |
| Ablageort | PublicationAdministration.java, PublicationAdministrationLocal.java | |
| Erfüllte Anforderungen |  | |
| 5.1.3.8. Conference Administration | | |
| Zweck | Änderungen an Konferenzen durchführen | |
| Schnittstellen | Bietet | Create conference |
| Cancel conference |
| Rate conference |
| Register to conference |
| Deregister from conference |
| Benötigt | Persist |
| Query |
| Ablageort | ConferenceAdministration.java, ConferenceAdministrationLocal.java | |
| Erfüllte Anforderungen |  | |

|  |  |  |
| --- | --- | --- |
| 5.1.3.9. Entities | | |
| Zweck | Persistente Speicherung der Daten | |
| Schnittstellen | Bietet | Persist |
| Named queries |
| Ablageort | Conference.java | |
| ConferenceRating.java | |
| ConferenceRatingPK.java | |
| ConferenceUserRole.java | |
| ConferenceUserRolePK.java | |
| Publication.java | |
| PublicationPK.java | |
| PublicationReview.java | |
| PublicationReviewPK.java | |
| PublicationRevision.java | |
| PublicationRevisionPK.java | |
| User.java | |
| Erfüllte Anforderungen | JPA basierte Persistenz | |

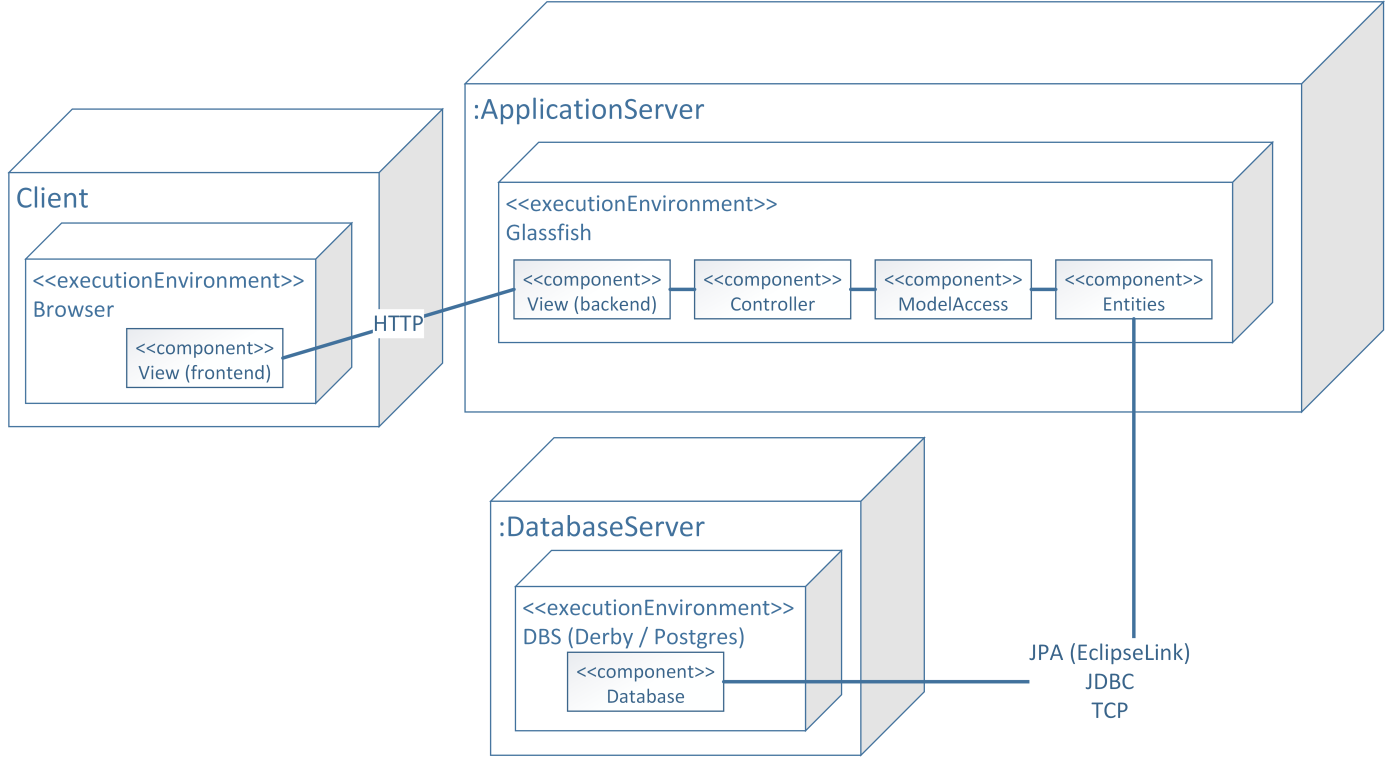
## Ebene 2

### Entitäten



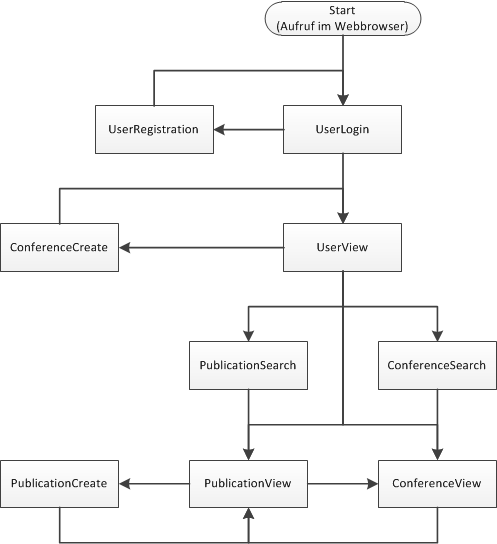
# Verteilungssicht

## Verteilungsdiagramm



# Konzepte

## Ablaufsteuerung (Page-Flow)



## Ausnahme- und Fehlerbehandlung

Fehler werden dem Benutzer aufbereitet als Growl-Notifications[[1]](#footnote-1) 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.

# Entwurfsentscheidungen

## 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.

## 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.

# Risiken

|  |  |  |
| --- | --- | --- |
| Priorität | Risiko | Erläuterung |
| 1 | Zeitmanagement | Einhaltung von Terminen |

1. Vgl. http://www.primefaces.org/showcase/ui/message/growl.xhtml [↑](#footnote-ref-1)