PRODYNA
information architects

05.07.2012

Leichtgewichtige Web 2.0-Architektur für komplexe Business-Anwendungen

Nicolas Moser
PRODYNA AG

Agenda

01 Einführung
02 Architektur
03 Lösungen
04 Zusammenfassung

Agenda

01 Einführung
02 Architektur
03 Lösungen
04 Zusammenfassung

Einführung

Architekturen für Business Anwendungen

- Was ist eine Business Anwendung?
  - Bildet verschiedene Prozesse eines Unternehmens ab
- Problematik
  - Immer wiederkehrende Anforderungen
  - Kein kompletter Ansatz für Business Anwendungen vorhanden
- NABUCCO Idee
  - Ein Framework mit einer Architektur für unterschiedliche Business Anwendungen zu schaffen

Einführung

Unsere Anforderungen an ein Framework für Business Anwendung

- Leichtgewichtig und Performant
  - Kurze Ladezeiten / Keine langen Wartepausen
  - Kontinuierliches Arbeiten ermöglichen
- Wiederverwendbar und Konfigurierbar
  - LEGO Baukastenprinzip
  - Business Anwendungen unterschiedlicher Arten
- Unabhängig
  - Open Source Framework
  - Keine Lizenzabhängigkeiten

Einführung

Unsere Anforderungen an ein Framework für Business Anwendung

- Web 2.0
  - HTML, CSS, JavaScript
  - AJAX, XHR, JSON
- Server-Zentrisch
  - Thin Client (beschränkt auf DOM Manipulation)
  - Ressourcen Orientiert (REST)
- KISS Prinzip

Einführung

Entwicklungsmöglichkeiten

- Zusammenstecken vorhandener Frameworks
  - Universeller Ansatz
  - Basisarchitektur
  - Getestete Funktionalität
  - Schnelle Ergebnisse
- Eigenentwicklung
  - Individueller Ansatz
  - Spezielle Lösungen für spezielle Probleme
  - Optimierungsmöglichkeiten
  - Keine Abhängigkeiten (Releases, Lizenzen)

Einführung

Heutige Web Anwendungen

- Beispiele für aktuelle Web Anwendungen
  - Amazon
  - Google Mail
  - Facebook
  - Wikipedia
  - Outlook Web Access

- Nicht auf Basis von Frameworks entwickelt

Agenda

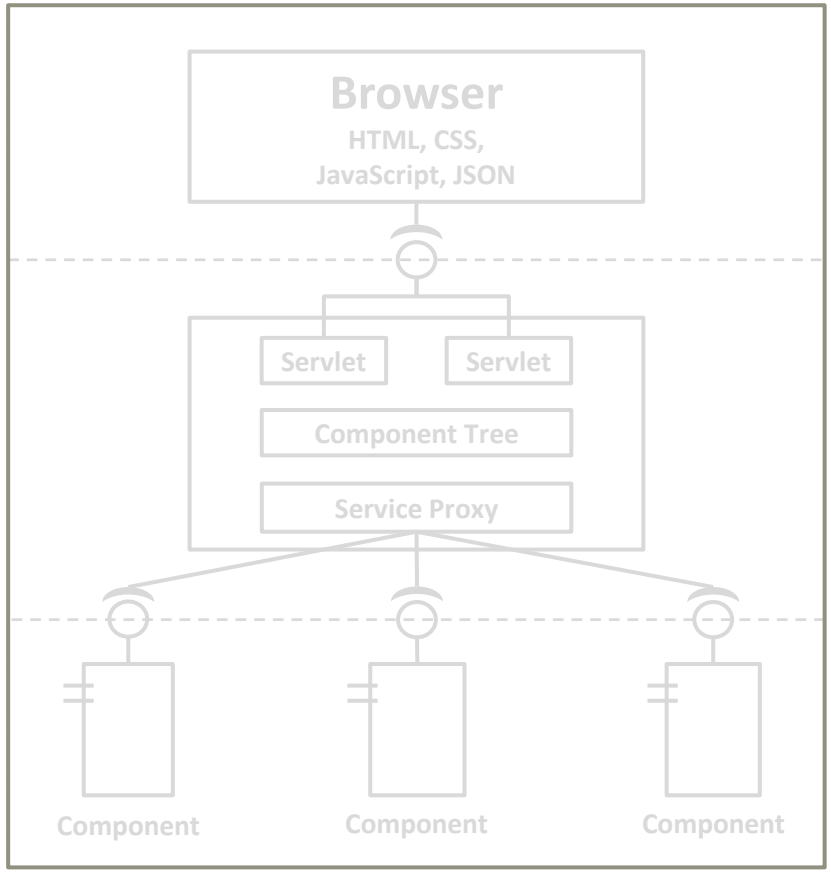
01 Einführung
02 Architektur
03 Lösungen
04 Zusammenfassung

Architektur

Client

Web Server

Application Server

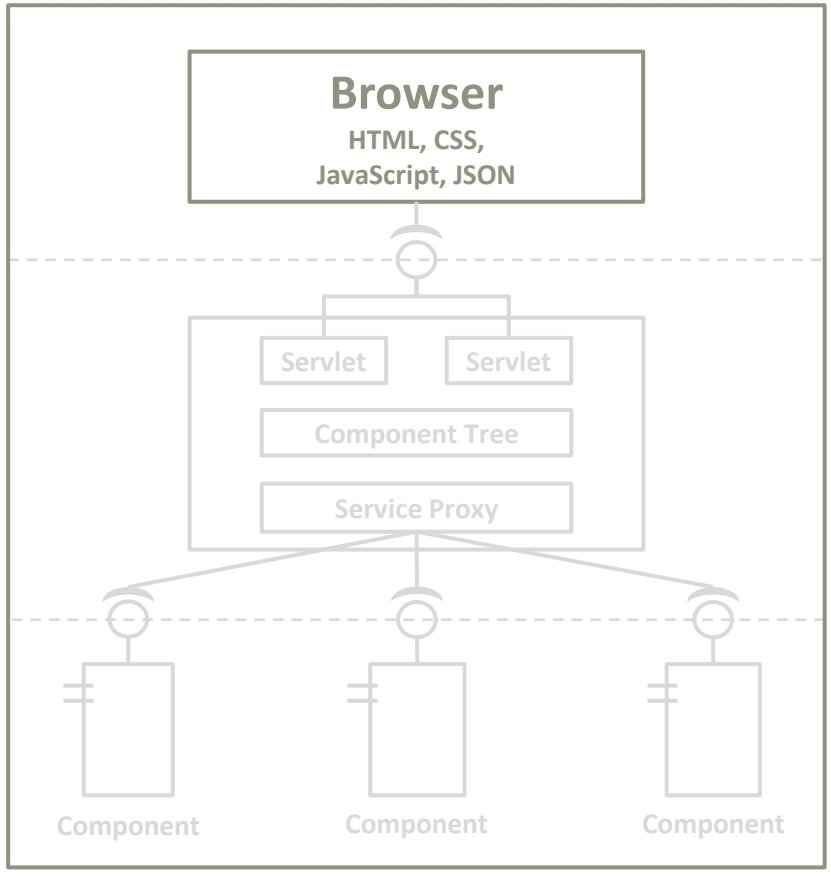


Architektur

Client

Web Server

Application Server

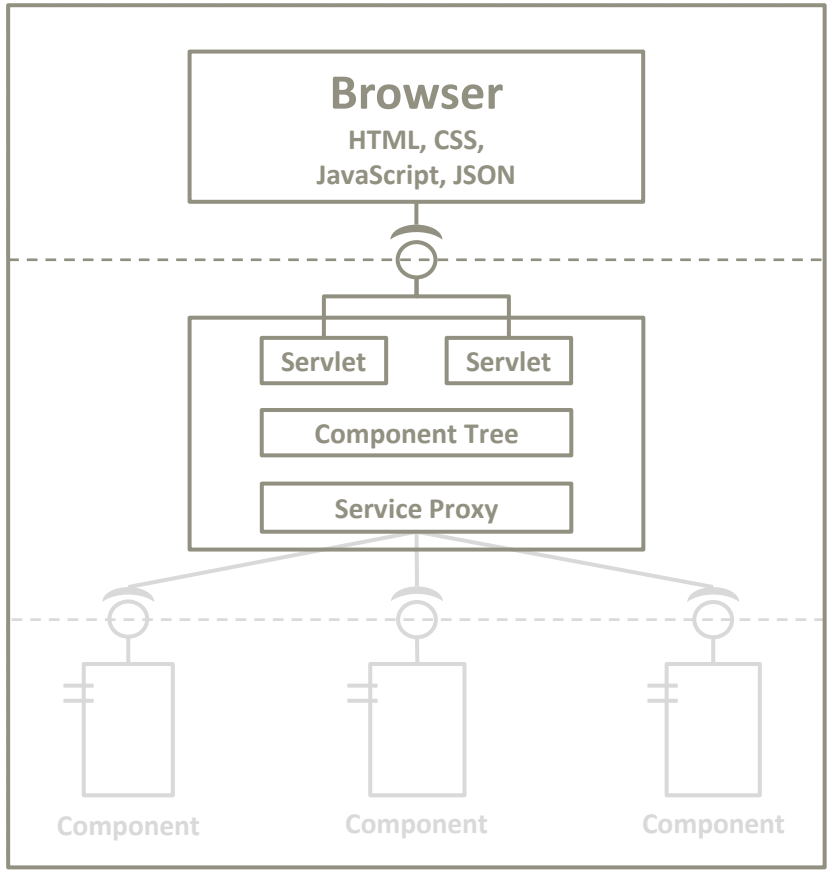


Architektur

Client

Web Server

Application Server



Database

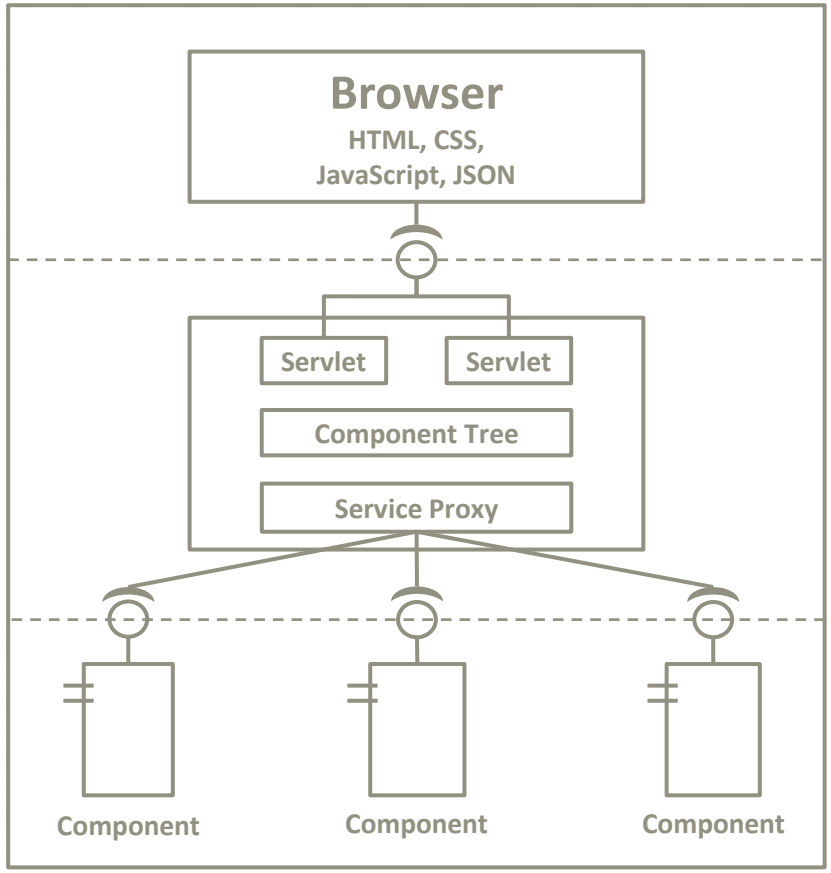
External System

Architektur

Client

Web Server

Application Server



Database

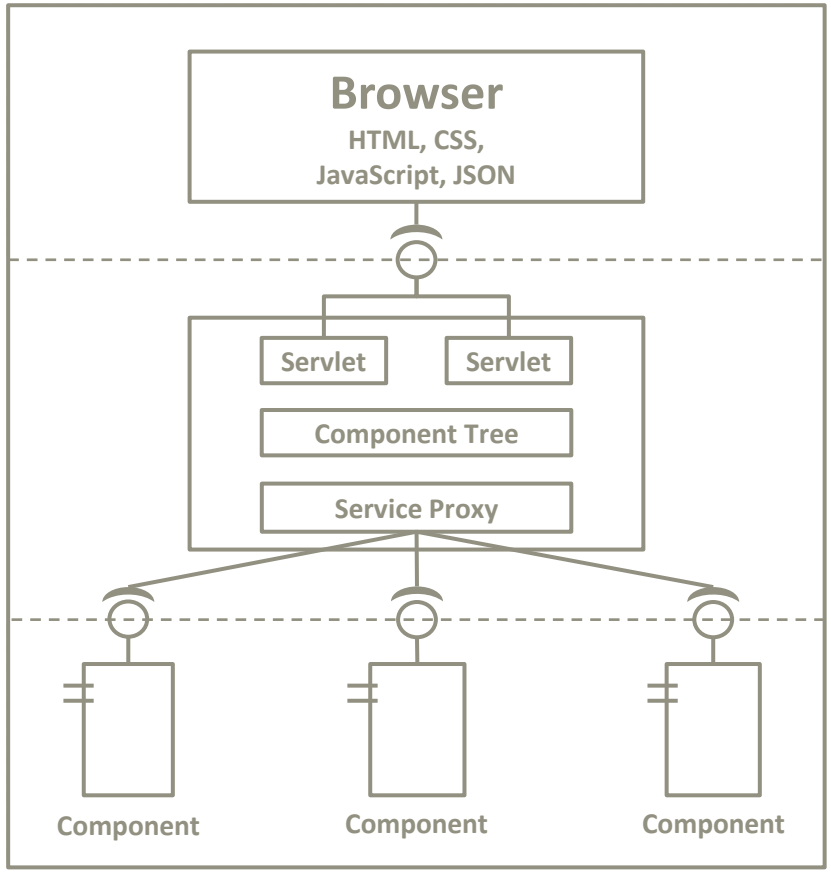
External System

Architektur

Client

Web Server

Application Server

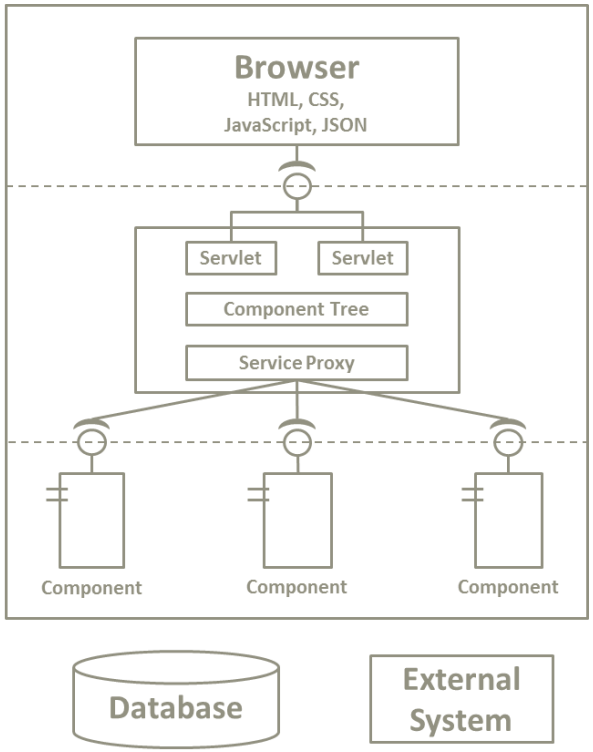


Agenda

01 Einführung
02 Architektur
03 Lösungen
04 Zusammenfassung

Lösungen

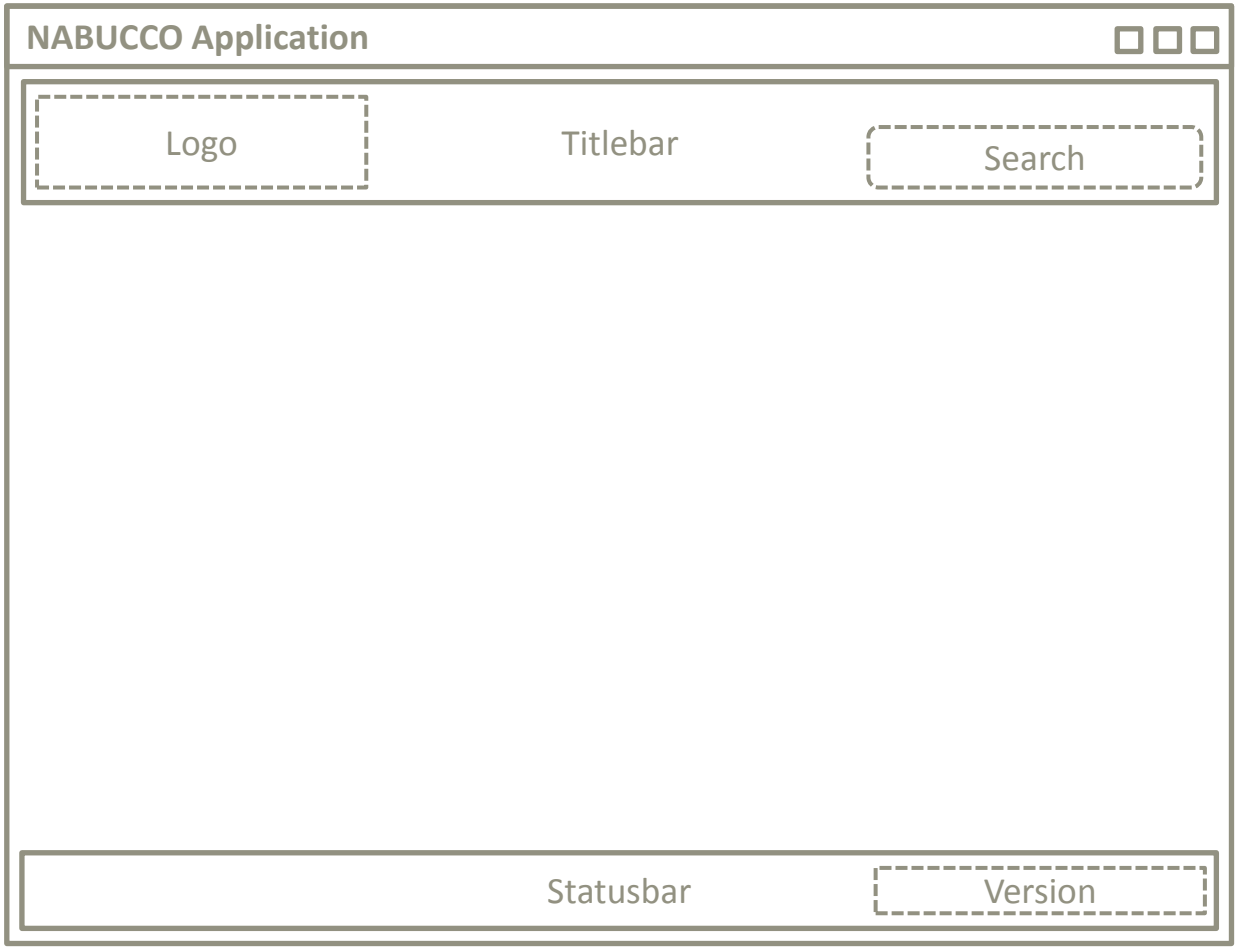
Layout



- Konfigurierbares Layout
  - Layout der Anwendung ist individuell konfigurierbar
  - Konfiguration per XML
  - Kein Anpassen von HTML, CSS, JavaScript notwendig
- Layouting nach LEGO Prinzip
  - Verschiedene Bestandteile
  - Individuell zusammensteckbar

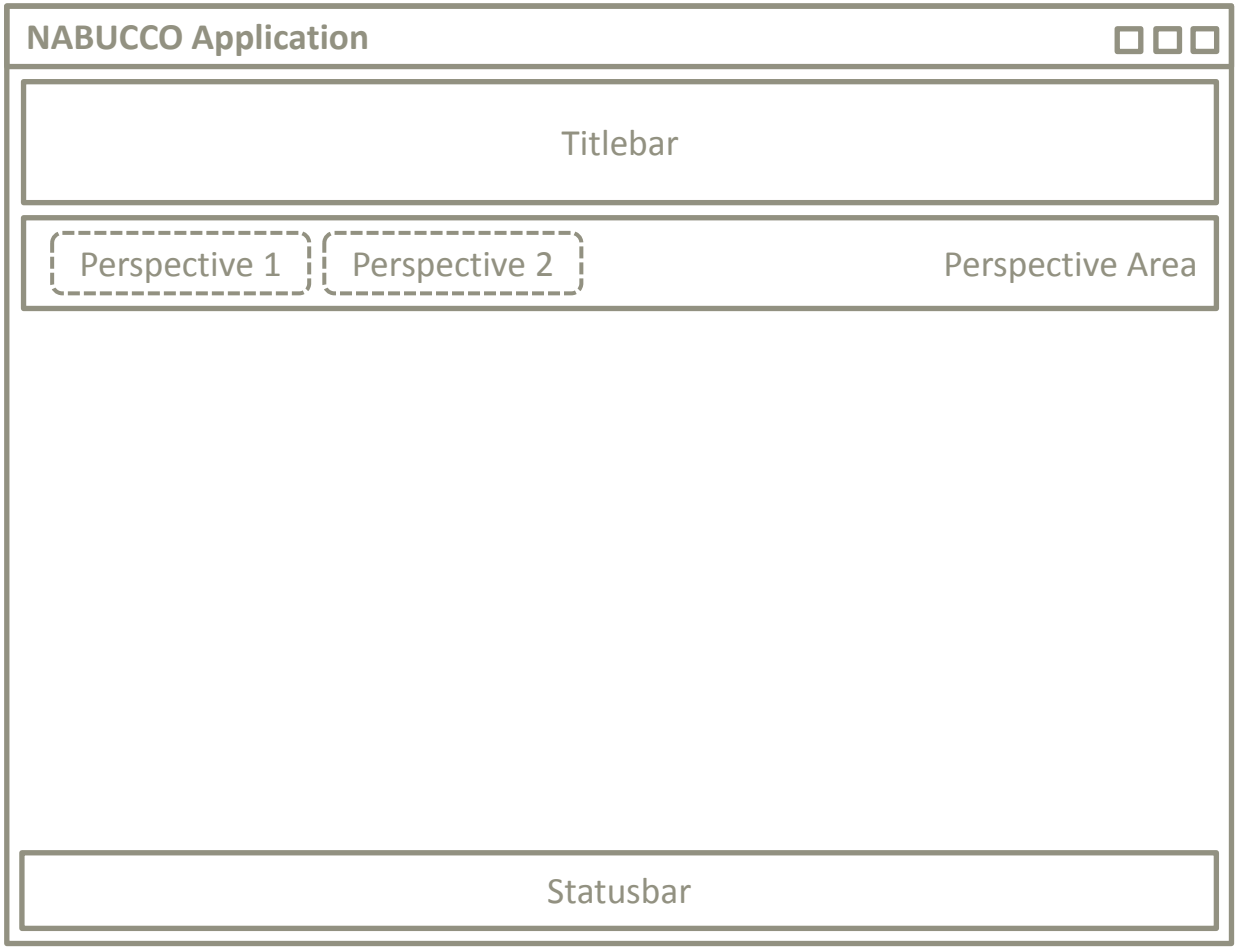
Lösungen

Layout



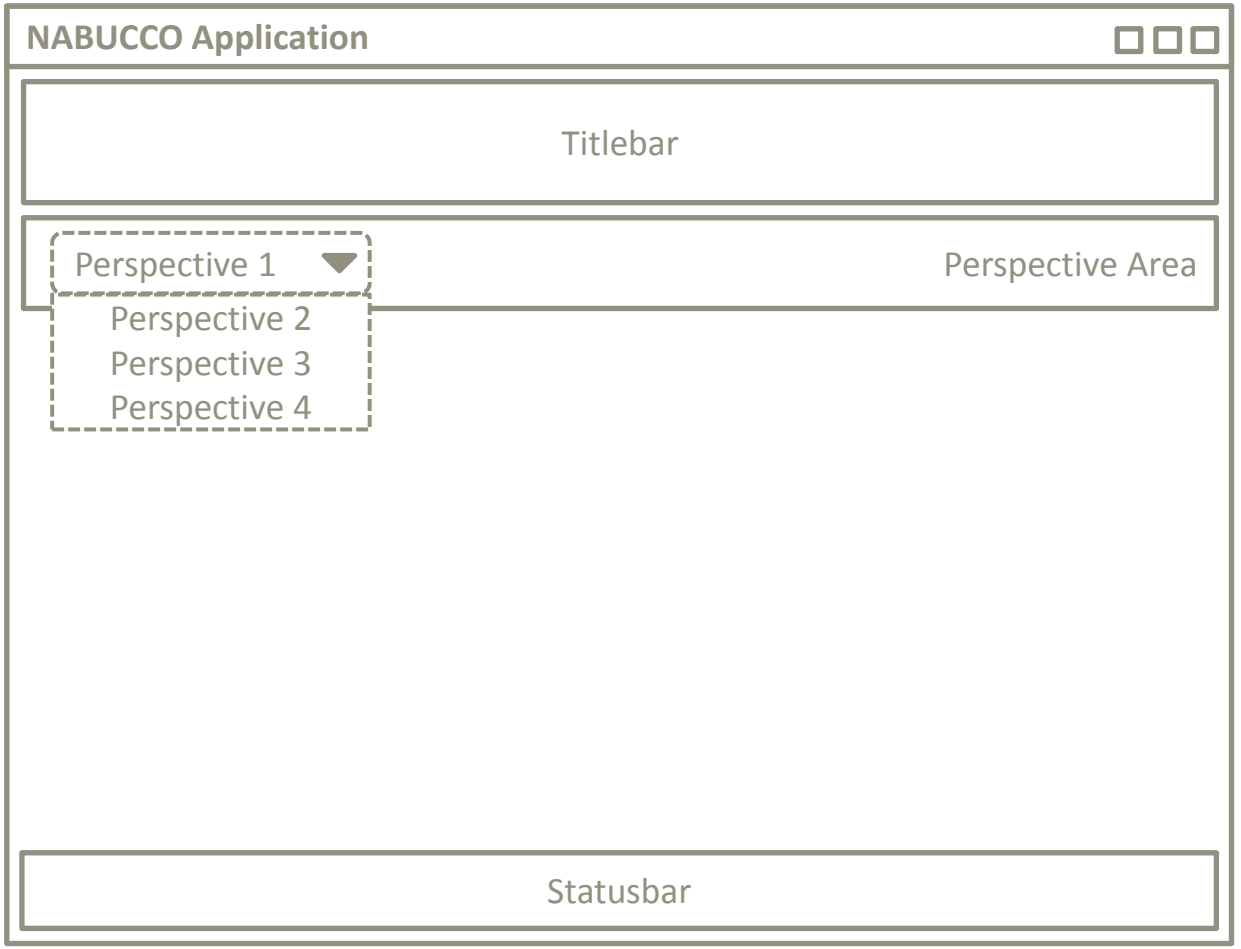
Lösungen

Layout



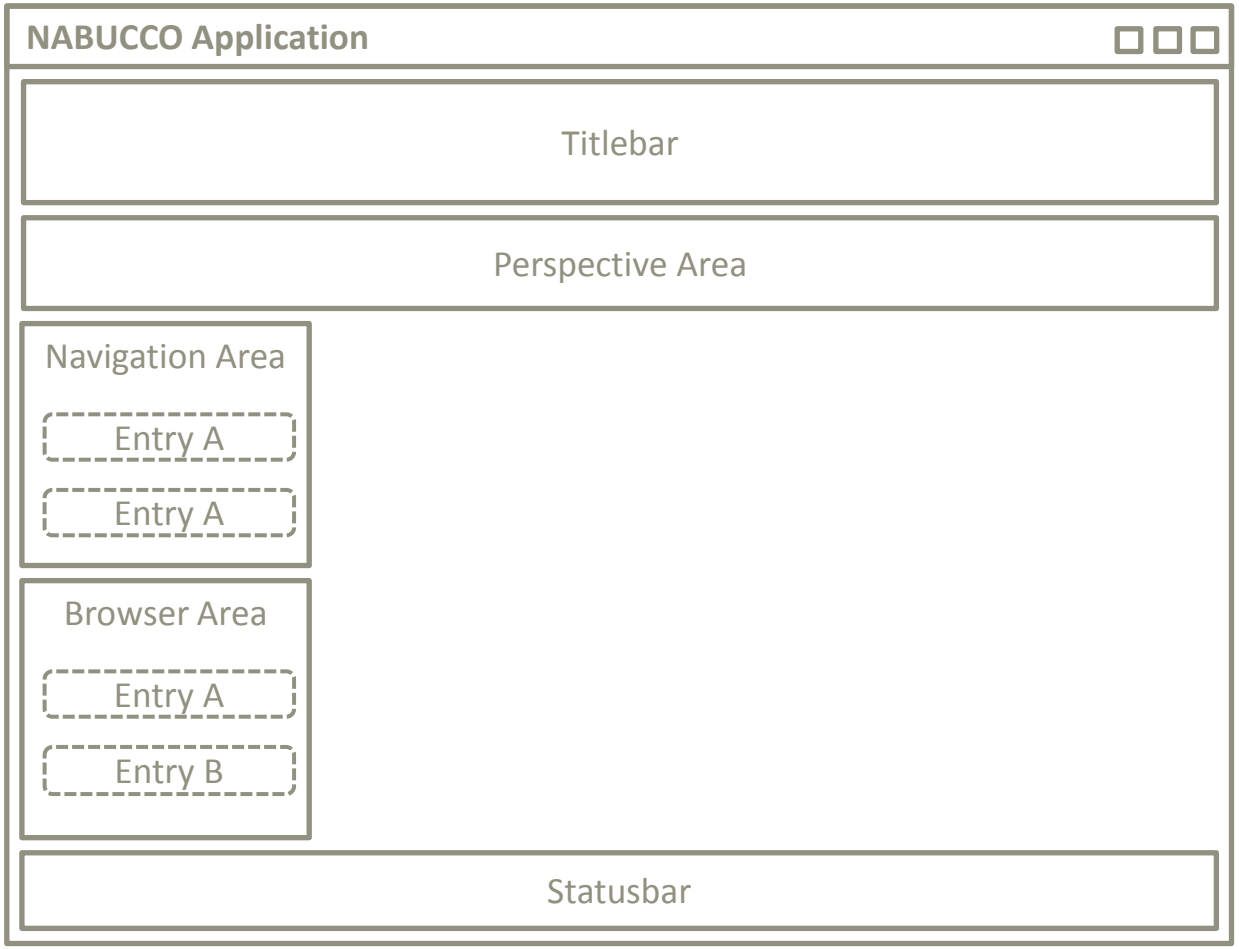
Lösungen

Layout



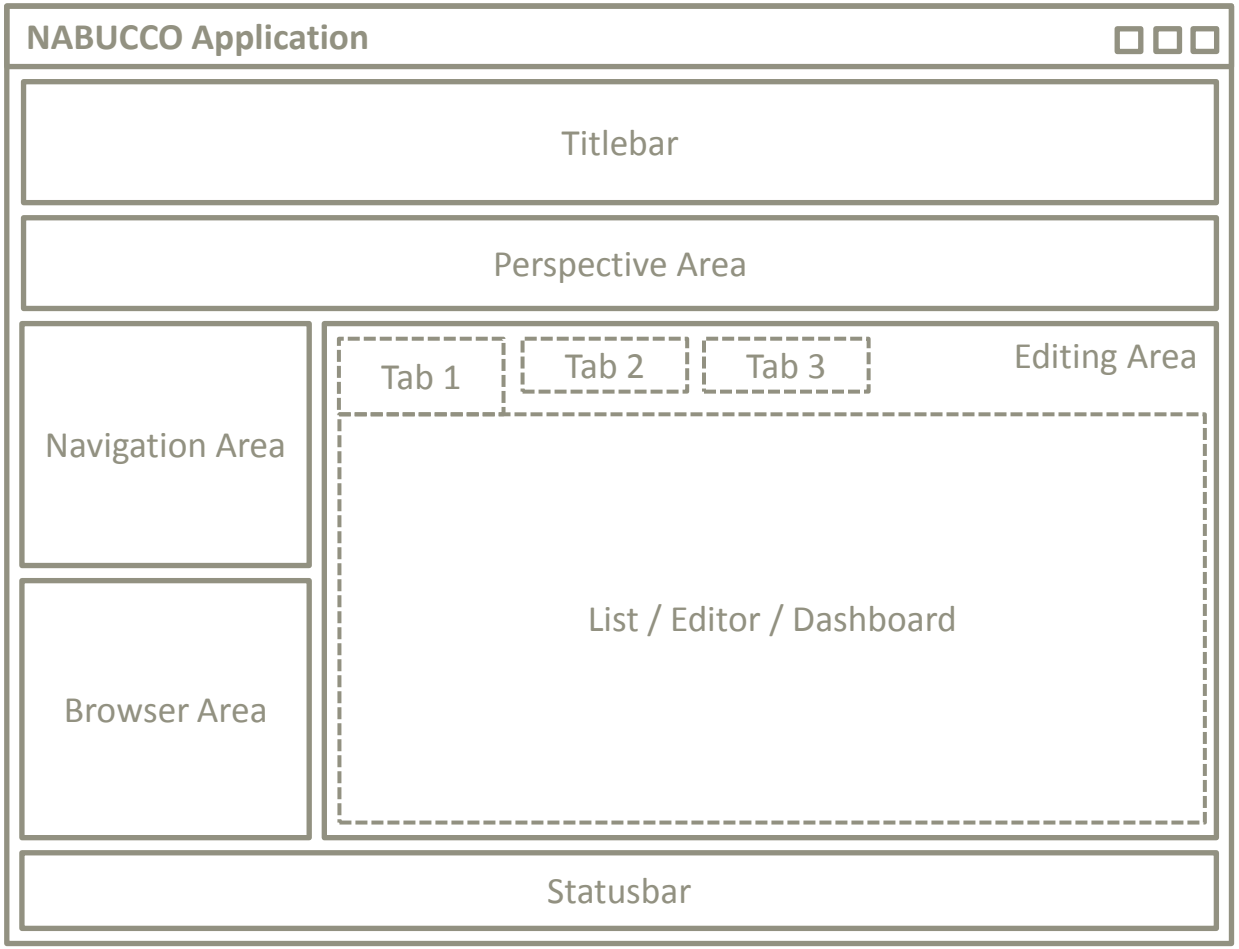
Lösungen

Layout



Lösungen

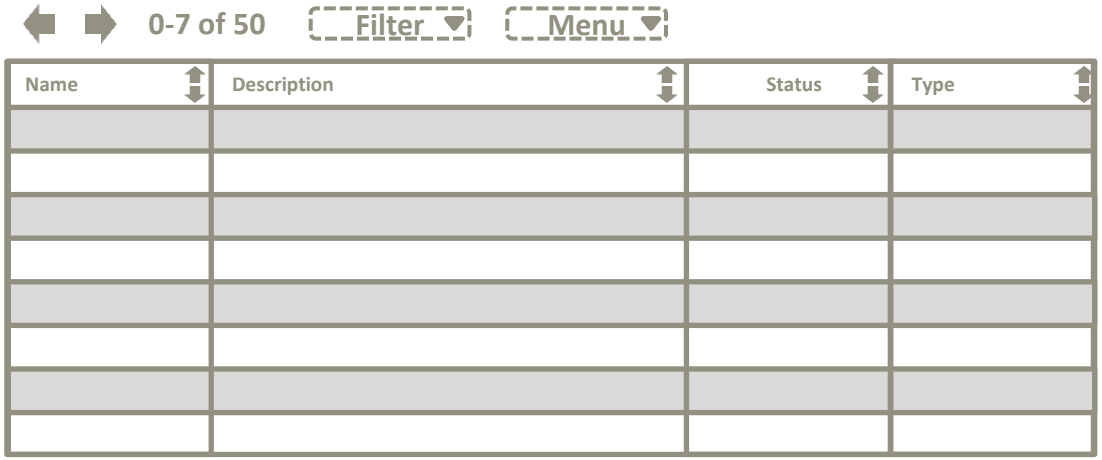
Layout



Lösungen

Layout

- Listen Layouting
  - Listen stellen Mengen von Daten in einer Tabelle dar
  - Serverseitiges Paging, Filtern & Sortieren
  - Konfigurierbare Titel, Größe, Filter, Sortierung




```
<column
    id="name"
    property="name"
    label="Name"
    tooltip="Projektname"
    width="15"
    visible="true"
    sortable="true" />
```

Column Konfiguration

Lösungen

Layout

- Editor und Dashboard Layouting
  - Aufteilung in zweidimensionales Grid
  - Controls werden auf dem Grid platziert
  - Layout Hints definieren die Ausrichtung

- Eigene Controls und Widgets
  - Text-Field, Combo-Box, Date-Picker, etc.
  - Bar-Chart, Pie-Chart, etc.

```
<text
    id="name"
    property="name"
    label="Name"
    tooltip="Projektname"
    position="A0-B0"
    hint="std" />
```

Text-Field Konfiguration

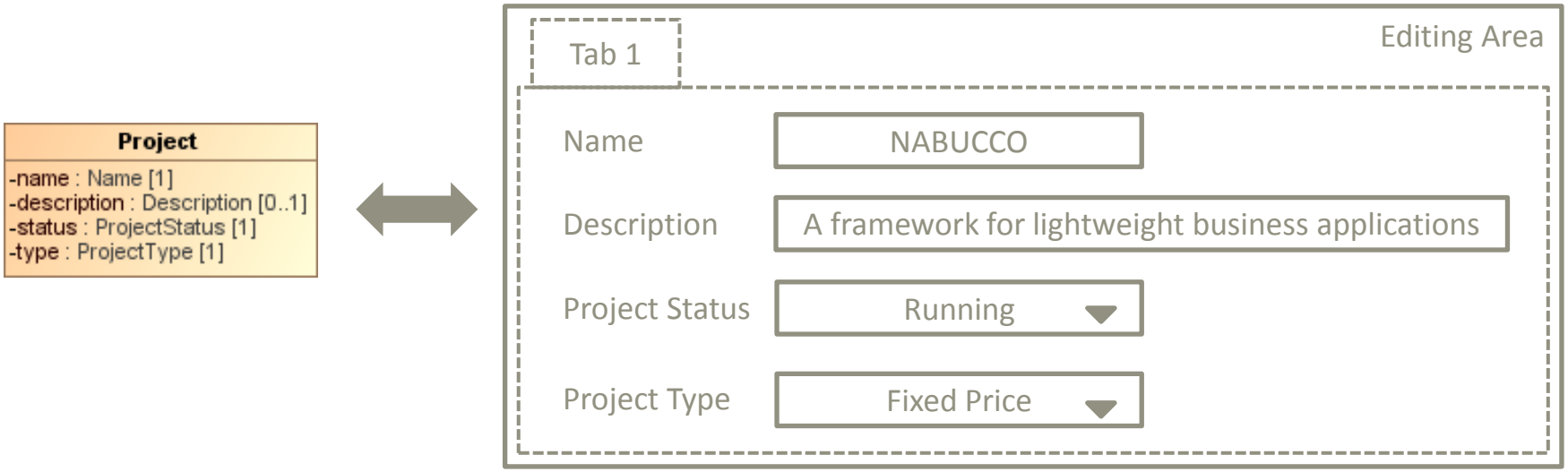
A0-B0

Name		C0	D0
A1	B1	C1	D1
A2	B2	C2	D2
A3	B3	C3	D3

Lösungen

Layout

- Binding
  - Controls werden per Property-Name an Attribute gebunden
  - Keine Verwendung von Java Reflection dank eigener MDA



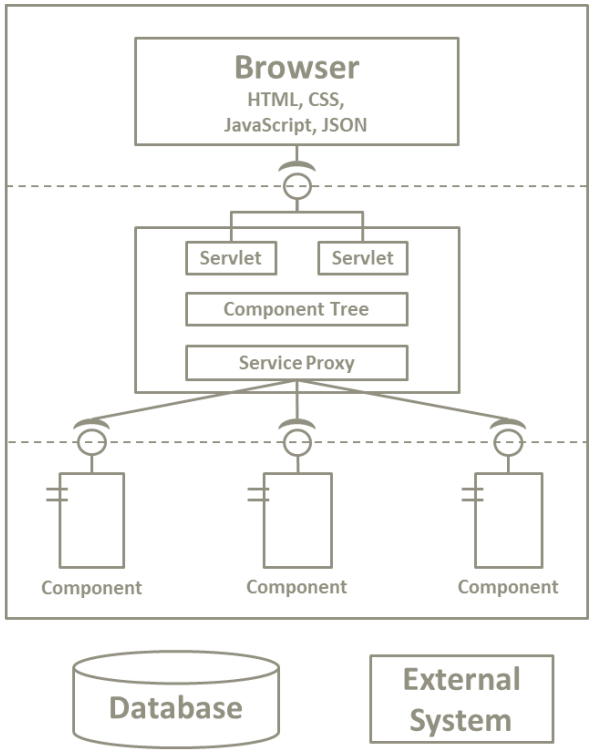
Lösungen

Live Demo

NABUCCO
Human Resources

Lösungen

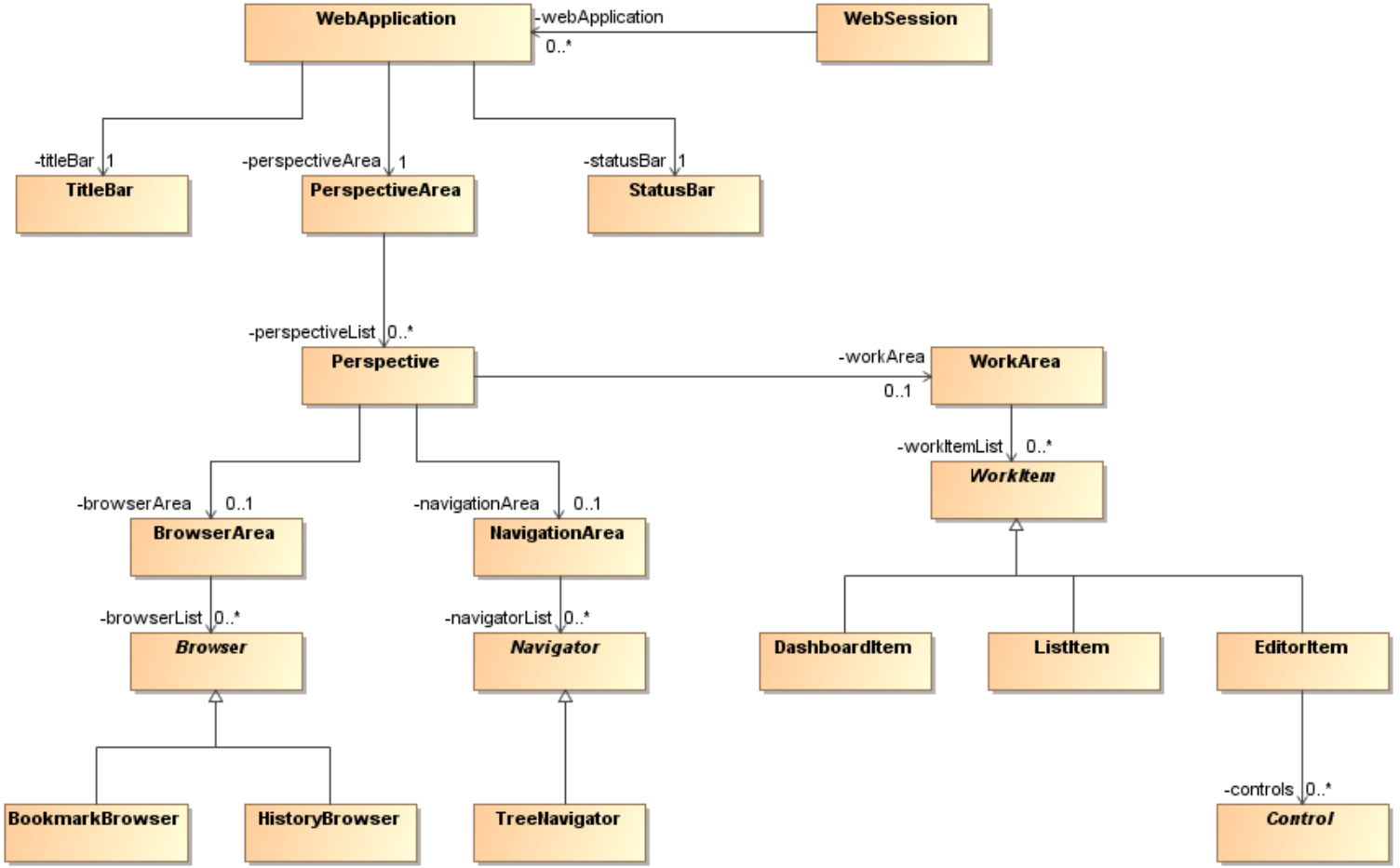
Komponentenbaum



- Server kennt und verwaltet den Client
  - Baum aller Anwendungsbestanteile
  - Serverseitige Abbildung der UI Elemente
- Ressourcensicht
  - Ressourcen werden per URL angesprochen
  - HTTP GET liefert die Ressource
  - HTTP PUT ändert die Ressource
  - HTTP POST legt neue Ressource an
  - HTTP DELETE löscht die Ressource

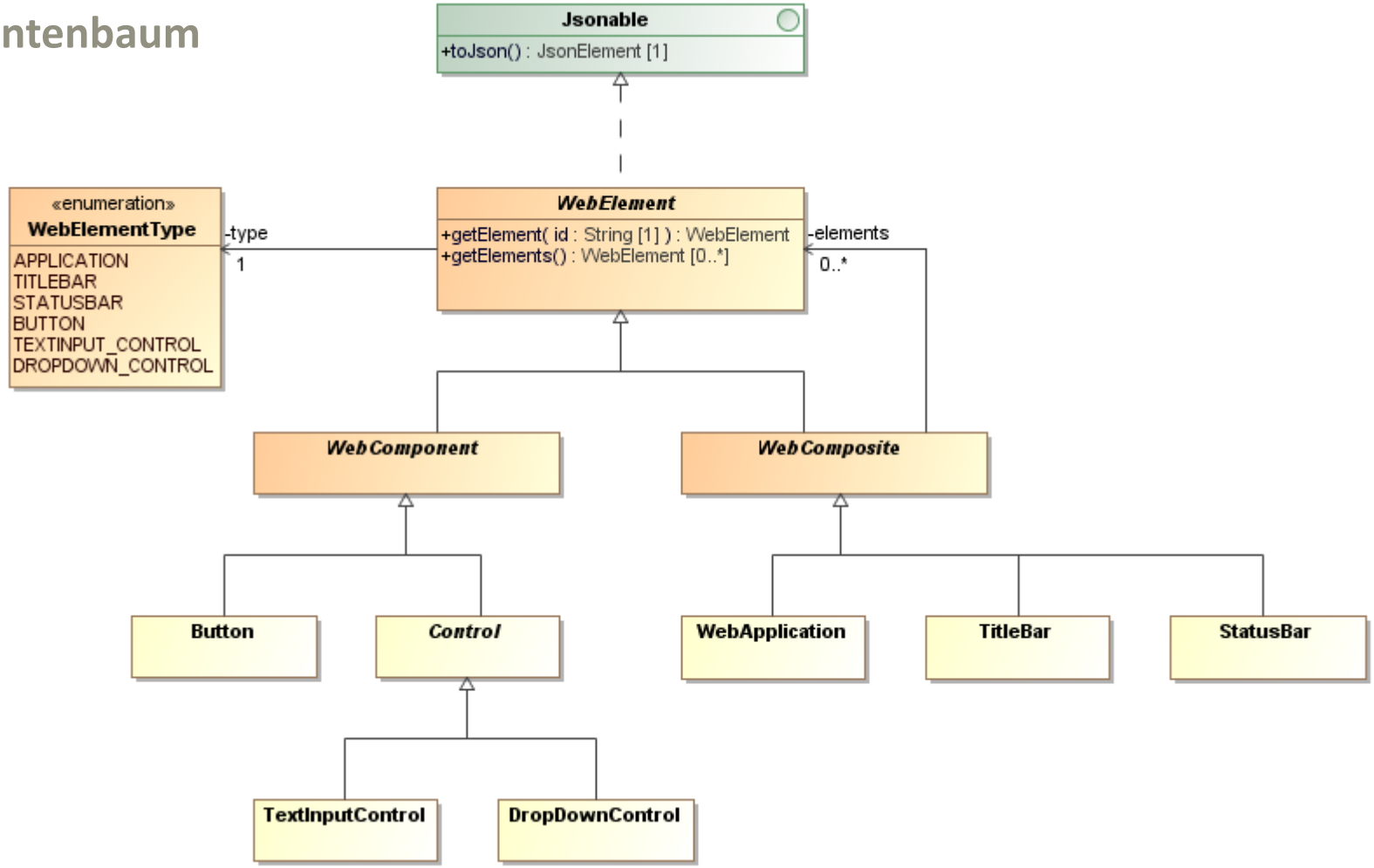
Lösungen

Komponentenbaum



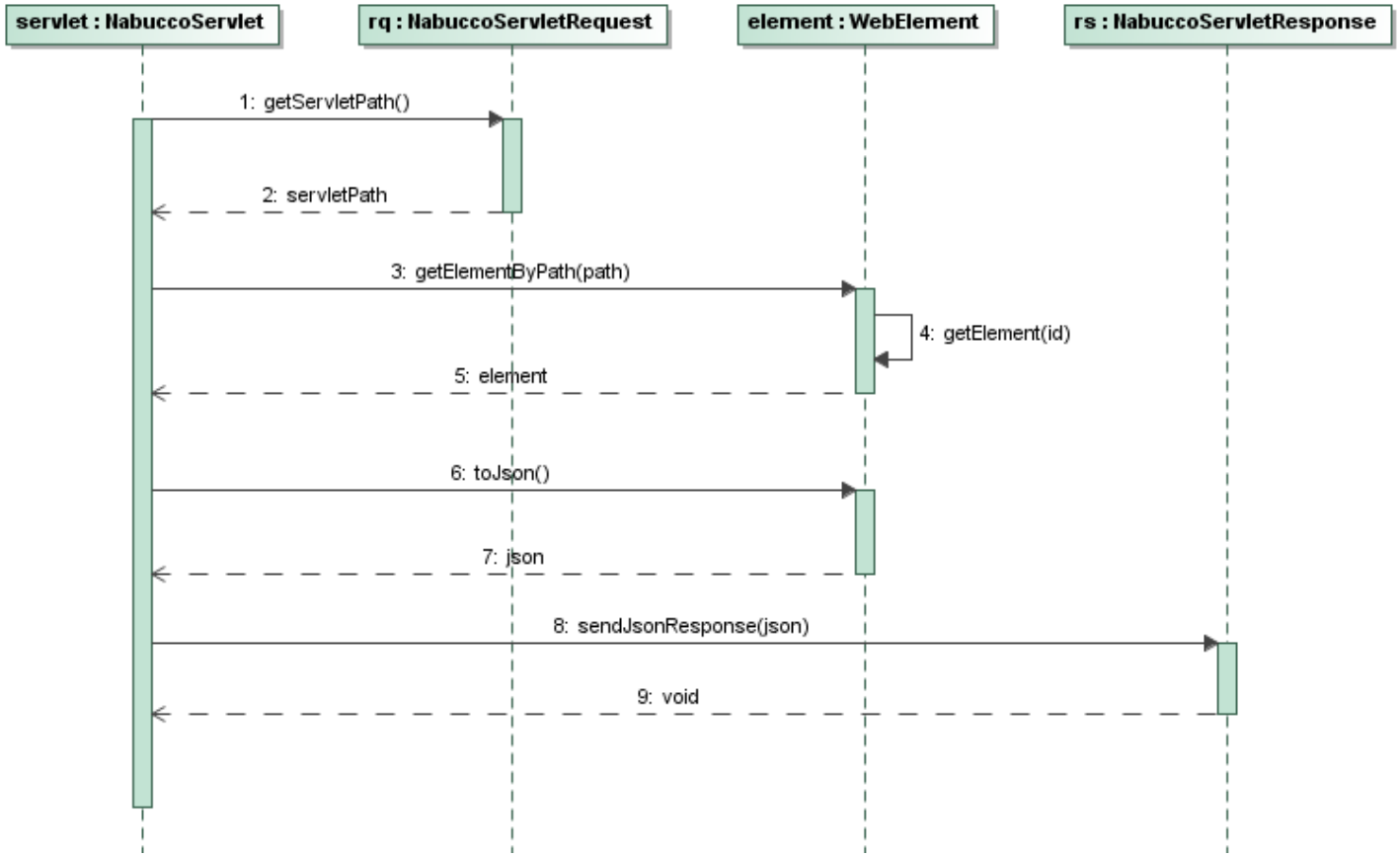
Lösungen

Komponentenbaum



Lösungen

Komponentenbaum



HTTP GET Request

Agenda

01 Einführung
02 Architektur
03 Lösungen
04 Zusammenfassung

Zusammenfassung

- NABUCCO Business Framework
  - Leichtgewichtig & Performant
  - Wiederverwendbar
  - Konfigurierbar
  - Simpel
  - Unabhängig
  - Open Source
- NABUCCO Human Resources
  - Referenzimplementierung

Fragen?

Uns gibt’s auch hier im Java Forum!





Stand Nummer 23

Vielen Dank für Ihre Aufmerksamkeit

- **NABUCCO**
  - http://nabucco.org/
  - http://github.com/nabucco/
- **PRODYNA**
  - http://www.prodyna.com/
  - http://www.prodyna.com/nabucco-business-framework/

Unsere Kontaktdaten

Ihr Ansprechpartner:

Nicolas Moser
Software Engineer
nicolas.moser@prodyna.de

PRODYNA AG
Ludwig-Erhard-Straße 12-14
65760 Eschborn
Telefon +49 69 597724-0
Telefax +49 69 597724-700
info@prodyna.com
www.prodyna.com

PRODYNA AG Stuttgart
Wilhelmsplatz 11
70182 Stuttgart
Telefon +49 711 596004-22
Telefax +49 711 596004-11