

Portale – Sicherheitsaspekte bei der Anwendungsintegration

Juri Urbainczyk

iteratec Architektur Workshop 2006



Themen der Anwendungsintegration

- Die Anwendung ist im Portal aufrufbar
- Anwendung und Portal bilden integrierte GUI
- Single-Sign-On (SSO)
- Datenübernahme nach Login
- Portal-Logout = Anwendungs-Logout
- Strukturintegration
- Autorisierungsdelegation
- Timeouts

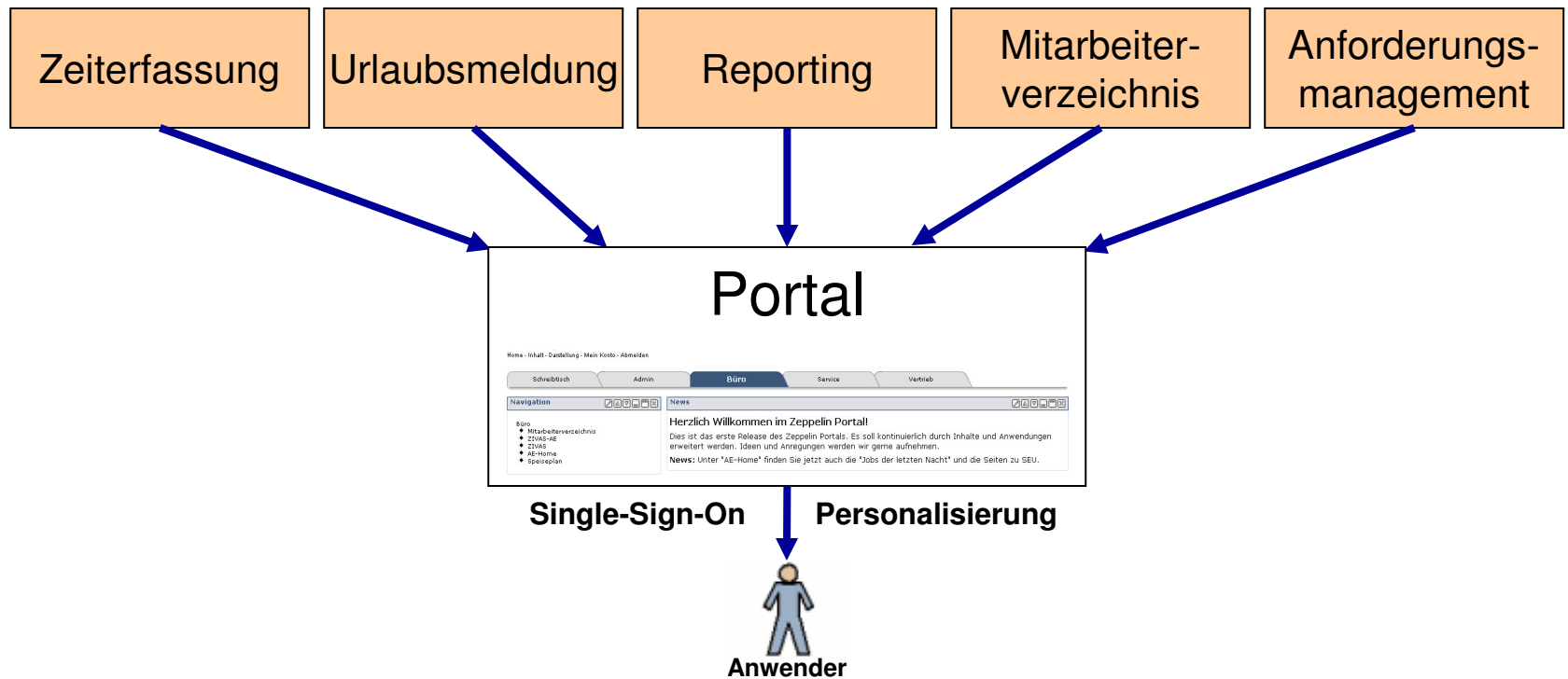
Agenda

- Projekthintergrund
- Aspekte & Grundprinzipien
- Authentisierung
- Autorisierung
- Timeout
- Voraussetzungen für zu integrierende Anwendungen

Hintergrund

- Projekt „Zeppelin Mitarbeiterportal“
- Februar – August 2006
- Basiert auf Liferay 3.6.1 – 4.1.2
- J2EE Opensource Portalserver
- JBoss 4.1 Applikationsserver unter Linux
- Ziel:
 - „Arbeitsplatz“ für die Mitarbeiter bereitstellen
 - Mittelfristig Integration aller Web-Anwendungen
 - Webseiten der Abteilungen ablösen
 - Single-Sign-On und Personalisierung
 - Infrastruktur für weitere Portalprojekte bereitstellen

Das Mitarbeiterportal



Die Einstiegsseite

Home - Abmelden

Büro IT Service Vertrieb

Navigation

Büro
ZIVAS
Urlaubsmeldung
Mitarbeiterverzeichnis
Zeppelin.de (link)

Ankündigungen

Herzlich Willkommen im Zeppelin Mitarbeiterportal!

Das Mitarbeiterportal integriert verschiedene Dienste und Informationen unter einem Dach.

Mit den oben stehenden Reitern (Tabs) und mit der Navigation auf der linken Seite gelangen Sie zu der Anwendung Ihrer Wahl.

Intranet Suche

Suchbegriff
Los

Intranet

Suchbegriff Los

Verzeichnisstruktur / Directory Structure

- ▶ Allgemeine Informationen
- ▶ Reisen
- ▶ Informationstechnik
- ▶ Unternehmenskommunikation
- ▶ Service Center Personal
- ▶ Finanzen
- ▶ Zentraleinkauf / Fuhrpark
- ▶ Bauen & Immobilien
- ▶ Vertrieb
- ▶ Service
- ▶ A180
- ▶ Schwarzes Brett
- ▶ Verkäufer-Wettbewerb "Southern Comfort"

Google Suche

Google™

Search

Portal Hilfe

Portal FAQ

Benutzer: zafio

Benutzerprofil
Rechte anzeigen

Dienste

Hilfe
erweiterte Suche
Dokumente

Verwaltung

Meine Dokumente
Admin Dokumente
Alle Dokumente
Schlagworte
Zugriffsstatistik

Grundprinzipien bei der Anwendungsintegration

- Schwache Kopplung zwischen Applikation und Portal
 - ungestörte, getrennte Weiterentwicklung
 - Option auf den Wechsel des Portalprodukts offen halten
- Wenig Änderungen am Portalprodukt
 - Migration auf zukünftige Versionen nicht gefährden
- Lösung darf nicht zu komplex sein
 - Administration des Portals soll handhabbar bleiben
- Eine doppelte Pflege von Daten ist zu vermeiden
- Der Aufwand für die Integration einer Anwendung muss vertretbar sein.
- Zukünftige Entwicklungen bei Liferay einbeziehen
 - z.B. rollenbasierte Navigation, mit LR 4 noch nicht möglich

URL Integration

Seiteneinstellungen

Private Seite bearbeiten: Zeppelin » Vertrieb » Reporting » GM-Reporting » Maschinenbestand

Seite Darstellung Import/Export

Übergeordnet: --- GM-Reporting

Name: Maschinenbestand Deutsch (Deutschland)

Art: URL

Versteckt: Nein

'Freundliche' URL: /group/zeppelin /vertrieb/reporting/gmreporting/ma

URL: http://reporting/qubon/server/cp/reports/maschinen_p/requests/t_Masc

Ziel: BLANK

Beschreibung:

Speichern Berechtigungen Löschen

URL Integration

The image shows a web application interface with a navigation menu on the left and a main content area on the right. The navigation menu includes sections for "Vertrieb", "Reporting", "Intranet Suche", and "Portal Hilfe". A black arrow points from the "Maschinenbestand" item in the "Reporting" section to the browser's address bar, which displays "http://reporting/". The main content area shows a form titled "Maschinenbestand" with various input fields and buttons.

Navigation Menu:

- Büro
- IT
- Service
- Vertrieb

Reporting Section:

- » GM-Reporting
 - Maschinenbestand
 - Replacementmgmt
 - Benchmarking Übersicht
 - Benchmarking Details
 - GM-Verkäufe
 - Ankaufsprämien
 - Replacementmgmt Upload
 - » VDMA-Reporting

Browser Address Bar: http://reporting/

Main Content Area:

>> Maschinenbestand << Organisationseinheiten

Wirtschaftsraum: (nichts ausgewählt)

Niederlassung: (nichts ausgewählt)

Auswahl des Verkäufers

Verkäufer: (nichts ausgewählt)

Verkäufernr.:

Auswahl des Kunden

Kundenr.:

Auswahl der Maschine

von Baujahr >= (Eingabe z.B. "1995"):

bis Baujahr <= (Eingabe z.B. "2000"):

Gerätetyp like:

Zur Hilfe | Zum Index | Abfrage starten

Integration mit IFrame - Beispiel

Home - Mein Konto - Abmelden - Inhalt hinzufügen - Seiten-Einstellungen - Meine Seiten > Zeppelin (Privat)

The screenshot displays a web application interface with a navigation bar at the top containing 'Büro', 'IT', 'Service', and 'Vertrieb'. Below the navigation bar, the main content area is an iFrame titled 'Order Tracking' containing the following elements:

- Order Tracking der Zeppelin IT** (Section Header)
- Search fields for **Anforderung Nr** and **Call Nr**, with an **Anfrage** button.
- Text: **Haben Sie keine Nr parat, können Sie sich mit einigen Auswahlkriterien eine Trefferliste erstellen, aus der Sie dann auswählen können.**
- Search criteria: **Stichwort (z.B. Hyster)**, **Ansprechpartner**, **Fachbereich**, and **Status**, each with a corresponding input field or dropdown menu.
- A **Suche** button.

On the left side of the interface, there is a sidebar with the following navigation links:

- IT
- Vorlaufkartenpflege
- Aktuelle Aufgabenliste
- Order Tracking**
- Produktionsjobs
- Releaseplanung
- MoveX/M3
- Monatsabschlüsse
- Bibliothek
- AE Team
- Informationen
- AE-Home (link)
- BS-Home (link)
- SO-Home (link)

Below the sidebar, there are two additional iFrames:

- Intranet Suche**: A search box with the label 'Suchbegriff' and a 'Los' button.
- Portal Hilfe**: A link to 'Portal FAQ'.

Integration mit IFrame

- IFrame-Portlet implementiert das HTML <IFRAME> Tag
- IFrame:
 - Virtueller Browser
 - Wird rein clientseitig im Browser abgearbeitet
 - Jeder IFrame hat einen URL-Parameter
 - Ergebnisse der URL Aufrufs werden im Frame angezeigt
- Für jede Anwendung wird (mindestens) eine Instanz des IFrame-Portlets benötigt

```
<IFRAME border="0" frameborder="NO" scrolling="NO"  
  SRC="http://intranet/suche_portal.php"  
  STYLE="BORDER:NONE; WIDTH:100%; HEIGHT:75px" >  
</IFRAME>
```

Konfiguration IFrame-Portlet

Büro IT Service Vertrieb

IFrame

Einstellungen Darstellung Berechtigungen

Quell-URL

Authentisieren

Art der Authentisierung

Methode

Benutzer-Name

Kennwort

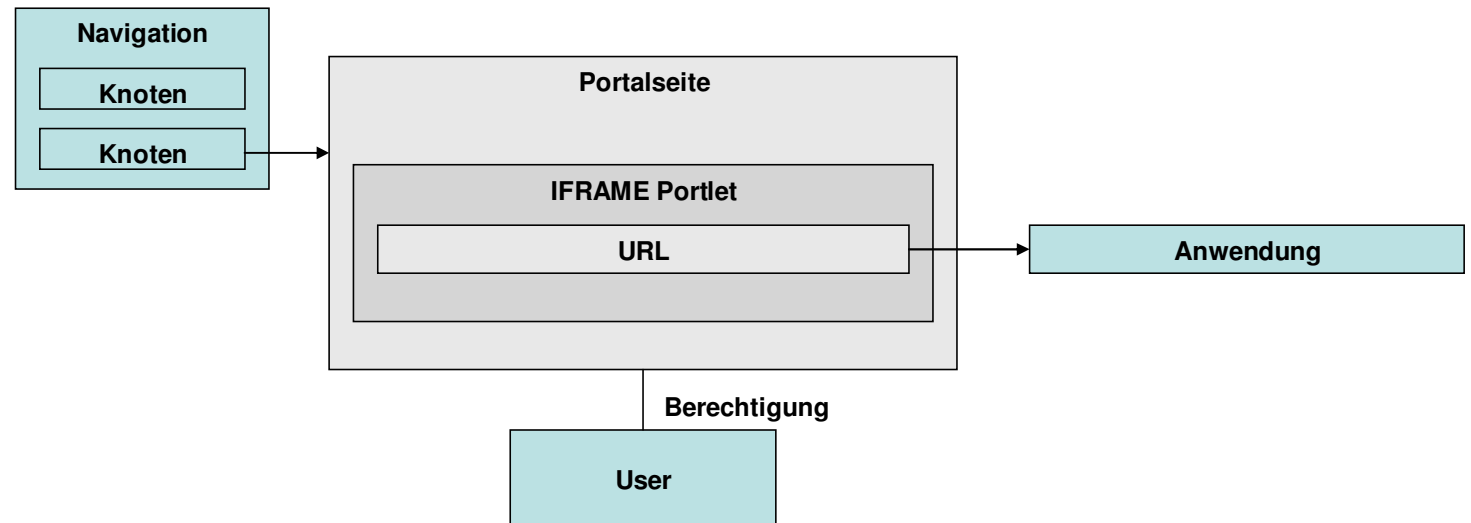
Versteckte Variablen

HTML Attribute

* Lassen Sie die Felder für Benutzername und -kennwort leer, um Ihre aktuellen Logindaten zu benutzen

Speichern

Anwendung und IFrame



Vor- und Nachteile

- Vorteile:

- Bei Änderungen am Portal ist die Anwendung nicht betroffen.
- Durch die Verwendung des IFrame wird der Portalserver nicht belastet (alle Zugriffe aus dem IFrame gehen direkt auf den Anwendungsserver).
- Es besteht keine direkte Kopplung zwischen Portalnavigation und Applikation. Dadurch können z.B. mehrere Anwendungen auf einer Portalseite vereinigt werden

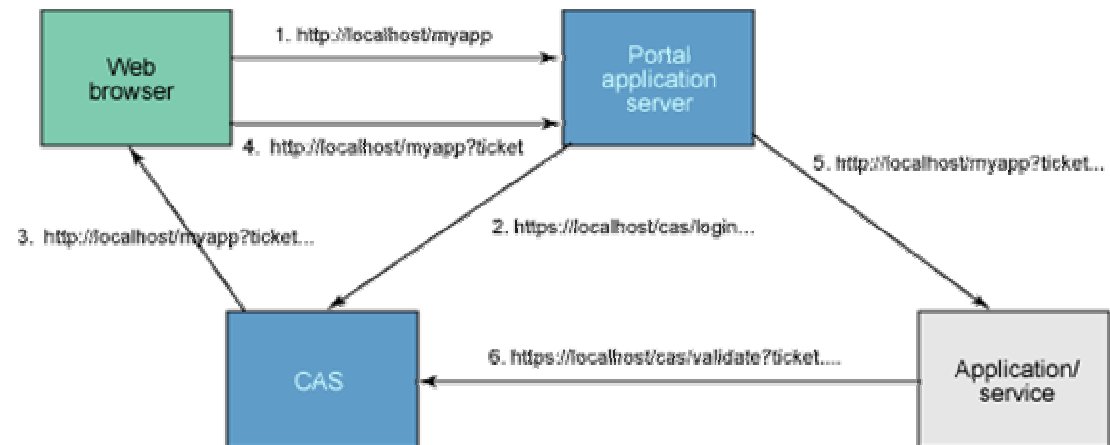
- Nachteile:

- Navigationsstatus der Anwendung kann verloren gehen.
- IFrame Implementierung unterscheidet sich zwischen verschiedenen Browsern (z.B. Zeichenkodierung).
- Anpassungen am Layout können trotzdem notwendig sein.

Single-Sign-On mit CAS

- Portale besitzen nur eingeschränkte SSO Funktionalität
- Kein SSO-Standard
- Opensource Software CAS = Central Authentication Service
- Integration in Liferay ist bereits vorbereitet
- CAS
 - Redirect, z.B. per Servlet Filter
 - Authentisierung per User/Password
 - Client-Implementierungen f. verschiedene Programmiersprachen vorhanden
 - Portal und Applikation sind gleichwertig (keine Reihenfolge)
- Integration von Web-Anwendungen mit wenig Aufwand

Funktionsweise von CAS

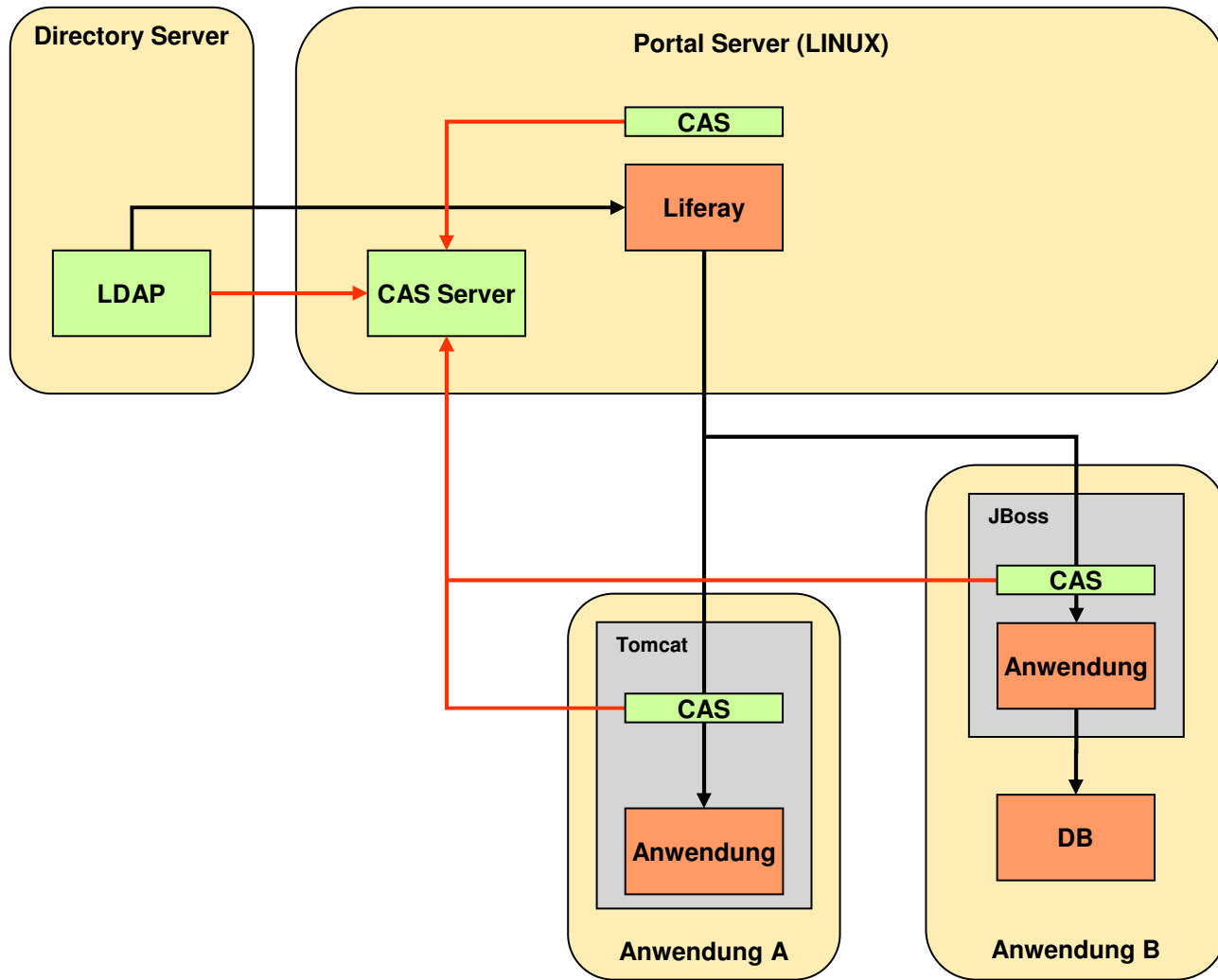


CAS Servlet Filter

```
<filter>
  <filter-name>CAS Filter</filter-name>
  <filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
    <param-value>https://ab.xy.corp/cas/login</param-value>
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>
    <param-value>https://ab.xy.corp/cas/serviceValidate</param-value>
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
    <param-value>reportingtest</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>CAS Filter</filter-name>
  <url-pattern>/server/cp/*</url-pattern>
</filter-mapping>
```

Architektur mit CAS



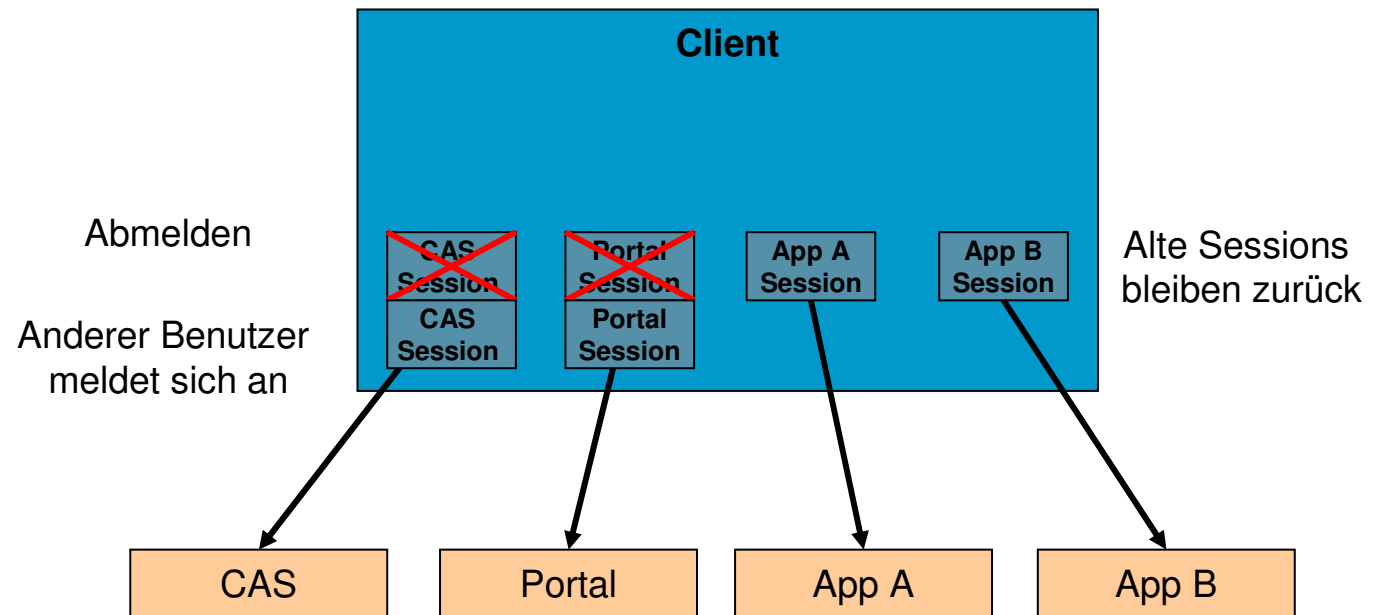
SSO - Konsequenzen

- Container-Security beachten
- Autorisierung bleibt Teil der Anwendung
 - Übernahme der Login-Informationen
 - Probleme mit Kaufsoftware
 - Alternativ: vorgeschaltete Autorisierung
- Alle Anwendungen sind abhängig von der SSO-Software
 - SSO-Software hat hohe Kritikalität
 - SSO-Konfiguration ist umgebungsabhängig
- Integration in mehrere Portale
 - Welches ist das „führende“ System?

Single-Sign-Off

- Portal-Abmeldung soll auch alle Sessions der Anwendungen beenden
- Sessions existieren nur bei den Anwendungen, die der Benutzer bereits aufgerufen hat
- Portal-Abmeldung beendet
 - ➔ Portal-Session
 - ➔ CAS-Session
- Wird der Browser nicht geschlossen, existieren die Sessions der integrierten Applikationen weiter und können von einem folgenden Benutzer verwendet werden
- Problem nur mit hohem Aufwand serverseitig lösbar

Single-Sign-Off



Single-Sign-Off

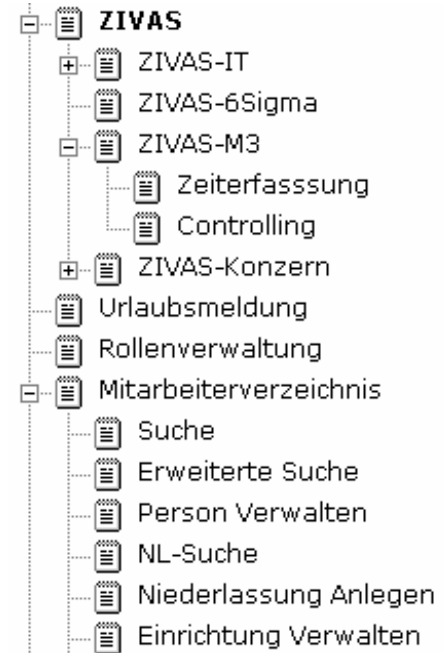
- Lösung:

- Nach Abmeldung erscheint die CAS „Logout-Seite“ (redirect)
- Pro integrierter Anwendung kommt ein unsichtbarer IFrame hinzu, der die Logout-URL der Anwendung aufruft.
- Diese Logout-URL löscht die aktuelle Session der betreffenden Anwendung.
- Bei der Anwendungsintegration muss auch die Logout-Seite gepflegt werden.
- Abhängigkeiten vom Browser beachten!

```
<DIV STYLE="visibility:hidden">  
    <IFRAME SRC="http://app1/logout"/>  
</DIV>  
<DIV STYLE="visibility:hidden">  
    <IFRAME SRC="http://app2/logout"/>  
</DIV>
```

Strukturintegration

- Anwendungen haben mehrere „Einstiegsunkte“ (=URLs)
- Die Einstiegsunkte repräsentieren unterschiedliche Anwendungsfälle
- Einstiegspunkte werden einzeln in das Portal integriert
- Das Menü der Anwendung wird somit in die Portalnavigation übernommen
- Engere Verzahnung zwischen Portal und Applikation



Autorisierungsdelegation

- Es sollen nur solche Einträge in der Navigation angeboten werden, für die der Benutzer berechtigt ist (Personalisierung)
- Das Portal benötigt also entsprechende Autorisierungs-Information
 - Das Wissen, welcher Benutzer für welchen Einstiegspunkt berechtigt ist, liegt in der Anwendung
 - Autorisierungsinformationen liegen in diversen Formen vor: hartkodiert, LDAP, Datenbank, textuell, ...
- Liferay bietet z.B. Berechtigungen auf Portalseiten
 - Seiten ohne Berechtigung werden automatisch nicht in der Navigation dargestellt
 - Diese Berechtigungen lassen sich nicht zu Rollen aggregieren

Seitenberechtigungen mit Liferay

Home

Seiteneinstellungen ←

Erlaubnis bearbeiten für: Seite: Home « Zurück

Benutzer Organisationen Standorte Benutzergruppen Verknüpft

zugeordnet vorhanden

Vorname Zweiter Vorname Nachname Email-Adresse

und Suchen

Berechtigungen aktualisieren

<input type="checkbox"/>	Name	Email-Adresse	Berechtigungen
<input type="checkbox"/>	Juri Urbainczyk	zafio@zeppelin.com	Aktualisieren, Anzeigen

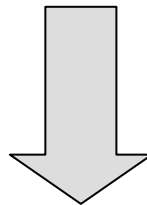
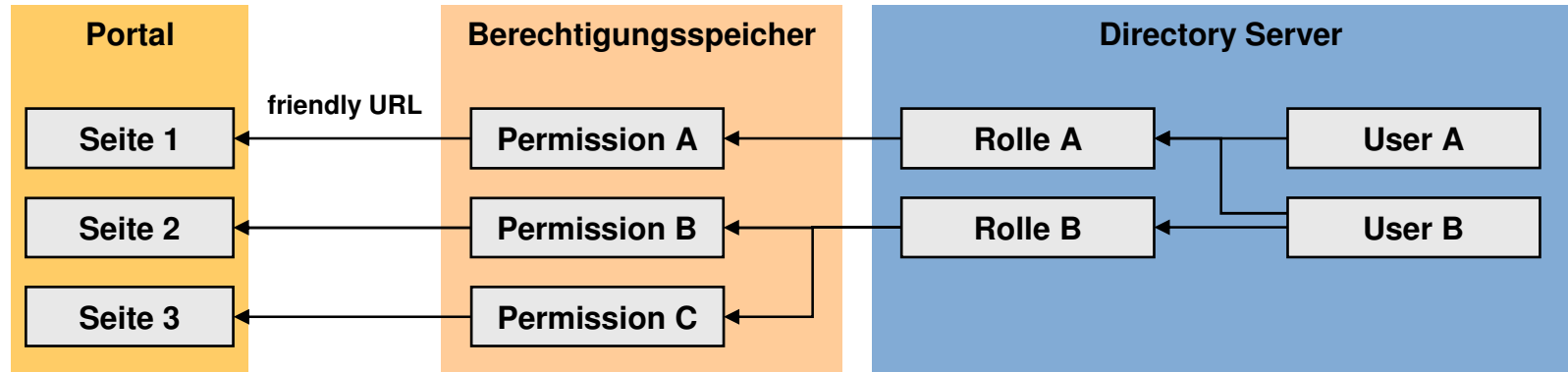
Autorisierungsdelegation – Konzept 1

- Vorschlag:
 - Schnittstelle, die jede Anwendung implementieren muss
 - Portal ruft Methoden der Schnittstelle beim Aufbau der Navigation
 - Methoden liefern Information, ob Navigationseintrag angezeigt werden soll, oder nicht
- Vorteil:
 - Autorisierungsinformation und Portal sind separiert
- Nachteile:
 - Schnittstelle läßt sich nicht immer zur Verfügung stellen (z.B. Kaufsoftware)
 - Unklar, wie Schnittstelle zu parametrieren wäre (z.B. mit aufzurufenden URL? Diese kann nur indirekt über Portlets ermittelt werden.)
 - Was ist bei statischen Inhalten?
 - Liferay Navigation muss neu implementiert werden!

Autorisierungsdelegation – Konzept 2

- Anwendungen müssen Autorisierungsinformation in Form von Rollen in einem LDAP-Directory hinterlegen.
- Mapping von Rollen zu Seitenberechtigungen
 - In einem separaten Berechtigungsspeicher
 - Seiten werden über „friendly URL“ referenziert
 - Rollen werden über Rollennamen angesprochen
- Auswertung beim Login
 - Alle Rollen des Nutzers werden gelesen
 - Über das Mapping werden die Berechtigungen bestimmt
 - Entsprechende Liferay Seitenberechtigungen werden angelegt
- Vorteile:
 - Nach erneutem Login wieder aktuell
 - Liferay-Mechanismen bleiben unverändert
 - Auch komplexe Mappings sind darstellbar

Autorisierungsdelegation



```
#if ($roles.contain("Rolle_B"))
    $permissions.putPagePermission("/seite2", "view")
    $permissions.putPagePermission("/seite3", "view, update")
#end
```

Timeout

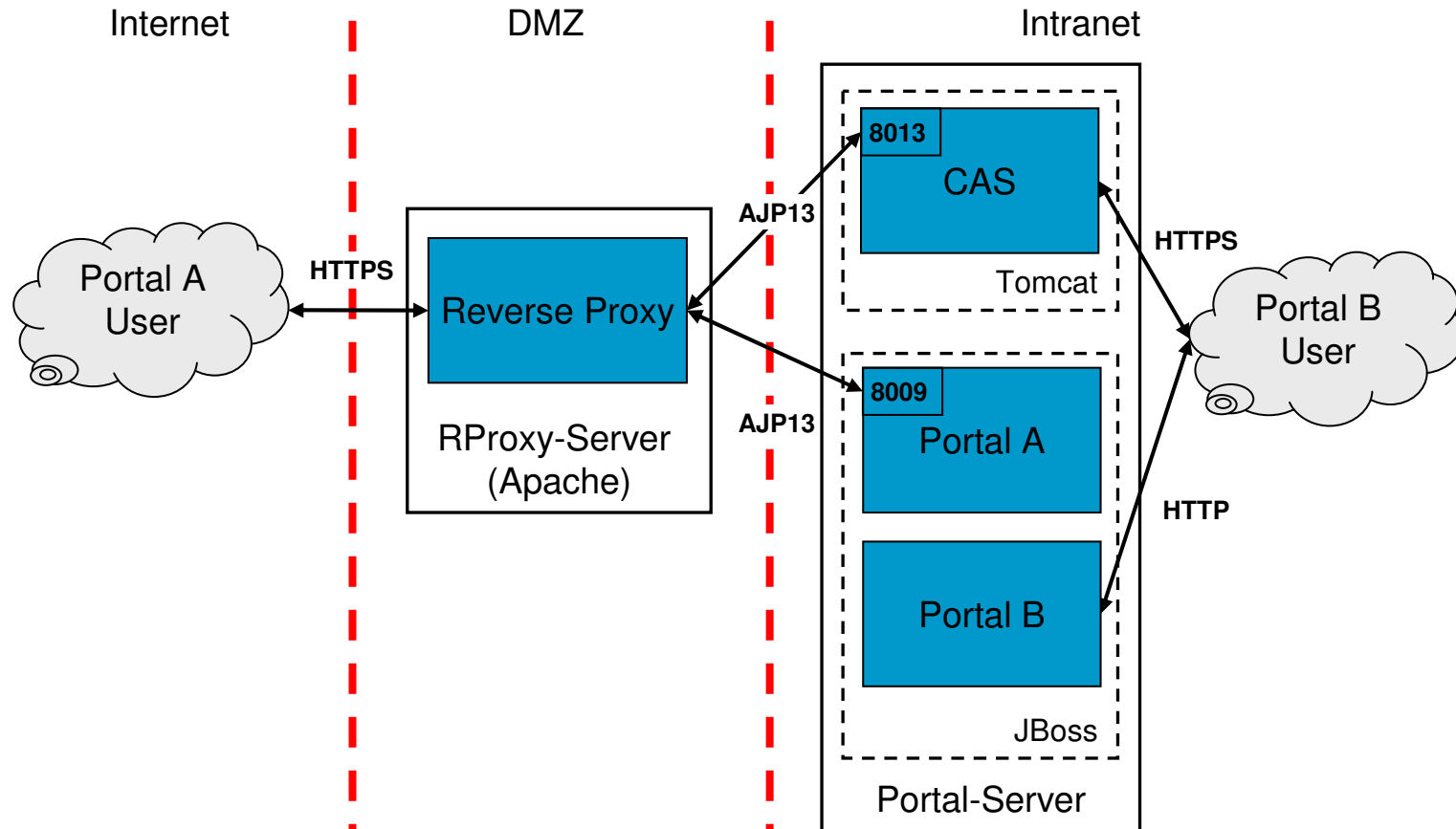
- Timeouts
 - ➔ Da nicht immer eine explizite Abmeldung stattfindet
 - ➔ Sessions würden sonst den Speicher des Servers füllen
- Portal-Timeout führt zu Datenverlust in Anwendungen
 - ➔ Portal-Session muss „erneuert“ werden, solange das Portal noch im Browser sichtbar ist

```
<% int sessionLength
    = ((request.getSession(true)).getMaxInactiveInterval() * 1000 ) - 30000; %>

<script type="text/Javascript" language="JavaScript">
<!--
var sessionLength=<%= sessionLength%>;
setInterval("sessionTimeout()", sessionLength);

function sessionTimeout() {
    loadPage("<%= themeDisplay.getPathMain() %>/portal/extend_session");
}
//-->
</script>
```

Mehrere Portale: Architektur



Betrieb mehrerer Portale

- Nutzung der Mandantenfähigkeit
 - Mehrere Portale innerhalb einer Applikationsserver-Instanz
 - Mehrere Portale innerhalb einer Liferay-Instanz durch Verwenden der *companyid*
 - Realisierung über Virtual Hosts
 - Umfang und Inhalt der Portal-Instanzen konfigurativ (Properties)
 - Eigene Benutzerverwaltung pro Portal-Instanz
- Vorteile:
 - Weniger Administrationsaufwand
 - ◆ Eine Applikationsserver-Instanz
 - ◆ Eine Datenbank-Instanz
 - Ein Build-Prozess
 - Eine Quellcode-Basis
 - Clusterfähig z.B. über JBoss Clustering

Konfiguration Virtual Hosts

server.xml (JBoss):

```
<Engine name="jboss.web" defaultHost="aportal.domain.de">
...
  <Host      name="bportal.domain.de"
            autoDeploy="false"
            deployOnStartup="false"
            deployXML="false">
    <DefaultContext
      override="true"
      crosscontext="true"
      cookies="true"/>
  </Host>
  <Host      name="aportal.domain.de"
            autoDeploy="false"
            deployOnStartup="false"
            deployXML="false">
    <DefaultContext
      override="true"
      crosscontext="true"
      cookies="true"/>
  </Host>
</Engine>
```

Die obigen Alias-Namen müssen im DNS-Server eingerichtet werden!

Voraussetzungen für zu integrierende Anwendungen

- Die Anwendung ist über eine (oder mehrere) URLs aufrufbar.
- Die Anwendung ist in einem IFrame sauber darstellbar.
- Die Anwendung kann den eigenen Login unterdrücken.
- Die Anwendung kann über den Aufruf einer URL ein Logout ausführen.
- Falls die Anwendung eine eigene Autorisierung durchführt, muss sie in der Lage sein, Login-Informationen aus der HTTP-Session entgegenzunehmen.
- Die Anwendung stellt notwendige Rollen im Directory Server (LDAP) zur Verfügung.
- Die Anwendung definiert ein Mapping von den Rollen zu den Einstiegspunkten (URLs).
- Ein Timeout darf nicht zu einem Popup führen.



Vielen Dank für Ihr Interesse.

Juri Urbainczyk

juri.urbainczyk@iteratec.de

