

# IV. CaptainCasa Community Meeting

# Agenda

- Vorstellung einzelner Anwendungsfälle
- Neue Features seit dem letzten Meeting
- Community Entwicklung: Services, Schulungen, ...
- Best Practices: Aha-Erlebnisse vor dem Produktivbetrieb
- Spring Framework
- Featureplanung

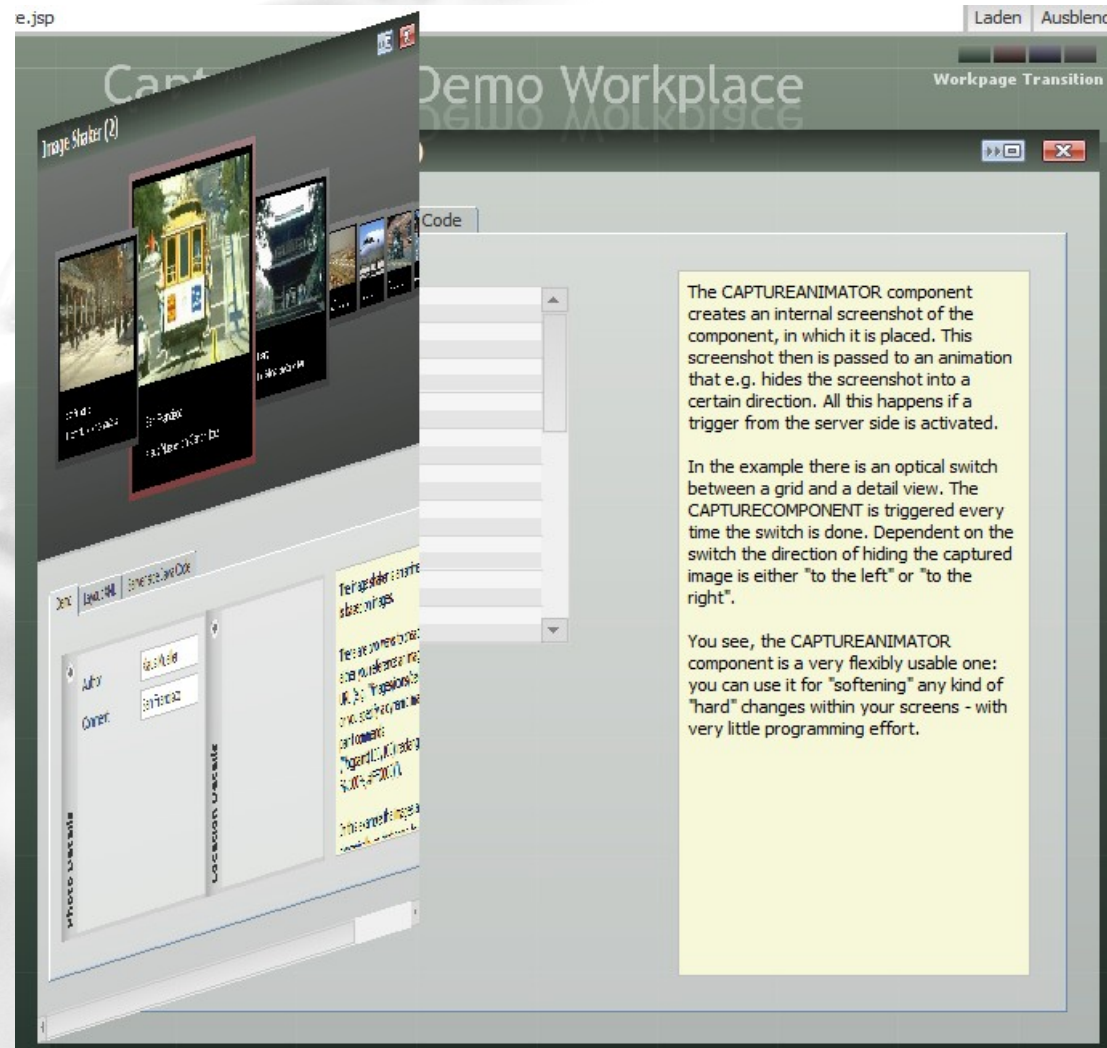


# Vorstellung einzelner Anwendungsfälle



# Neuerungen seit dem letzten Community Meeting

# Animationen - CAPTUREANIMATOR



The screenshot shows a web application interface with a 'Demo Workplace' and a 'Workpage Transition' window. The transition window displays a grid of images and a 'Code' panel with text explaining the CAPTUREANIMATOR component.

The CAPTUREANIMATOR component creates an internal screenshot of the component, in which it is placed. This screenshot is then passed to an animation that e.g. hides the screenshot into a certain direction. All this happens if a trigger from the server side is activated.

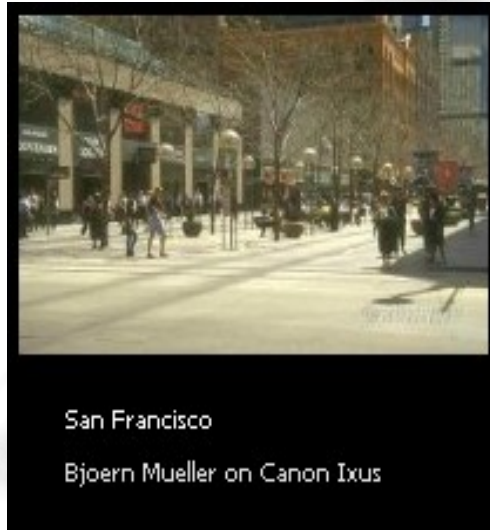
In the example there is an optical switch between a grid and a detail view. The CAPTURECOMPONENT is triggered every time the switch is done. Dependent on the switch the direction of hiding the captured image is either "to the left" or "to the right".

You see, the CAPTUREANIMATOR component is a very flexibly usable one: you can use it for "softening" any kind of "hard" changes within your screens - with very little programming effort.

# Animationen - IMAGESHAKER



# Styling - Dynamisches IMAGE



```
image = „/bgpaint(100,100):command1();command2();...“
```

# Styling - BGPAINT



```
rectangle(0,0,100%,100%,#680000,#F0F0F0,horizontal);  
write(10,10>Hello,lefttop);  
write(10,40>Hello,20,#FFFFFF80,lefttop);  
write(140,20,Some text,20,#FFFFFF80,lefttop,0);  
write(140,60,Some text,20,#FFFFFF20,lefttop,0,1,-1);  
write(100,100%-5,Some text,20,#FFFFFF80,lefttop,100);  
write(100%-25,5>Hello,20,#00000080,righttop,90)
```



# Grids - Variable header/footer heights

First Name	Last Name								
First Name - second headline	Last Name - second headline								
FN 0	LN 0								
FN 1	LN 1								
FN 2	LN 2								
FN 3	LN 3								
FN 4	LN 4								
First Name - footer line	Last Name - footer li								
			Staff	Office	IT	Other	Leisure	Pleasure	Company Cars
		▷ Board	0	0	0	0	0	0	0
		▷ Production	0	0	0	0	0	0	0
		▷ Sales	0	0	0	0	0	0	0
		▷ Development	0	0	0	0	0	0	0
		▫ CC 3.0	0	0	0	0	0	0	0
		▫ CC 3.1	0	0	0	0	0	0	0
		▫ CC 3.2	0	0	0	0	0	0	0
		▫ CC 3.3	0	0	0	0	0	0	0
		▫ CC 3.4	0	0	0	0	0	0	0
		▫ CC 3.5	0	0	0	0	0	0	0
		▫ CC 3.6	0	0	0	0	0	0	0
		▫ CC 3.7	0	0	0	0	0	0	0
		▷ Marketing	0	0	0	0	0	0	0

## Grid - FIXGRID-ROW\* Attributes

- ROWBGPAINT
- ROWPOPUPMENU
- ROWDRAGSEND
- ROWDROPRECEIVE
  
- No necessity to define e.g. drag&drop for each grid cell component anymore...

# Grid - Server Side Roundtrips

- **FIXGRID-AVOIDROUNTRIPS**
  - No roundtrip „in general“ when navigating through grid rows
  - Roundtrip when changing data within a row

## Grid - ARRAYGRID

- Fast and „slim“ variant for data output grids
- Much smaller memory Consumption on Server Side
- Faster Data Collection on Server Side
- ...not as flexible as FIXGRIDs.

# Grid / Tree Keyboard Navigation

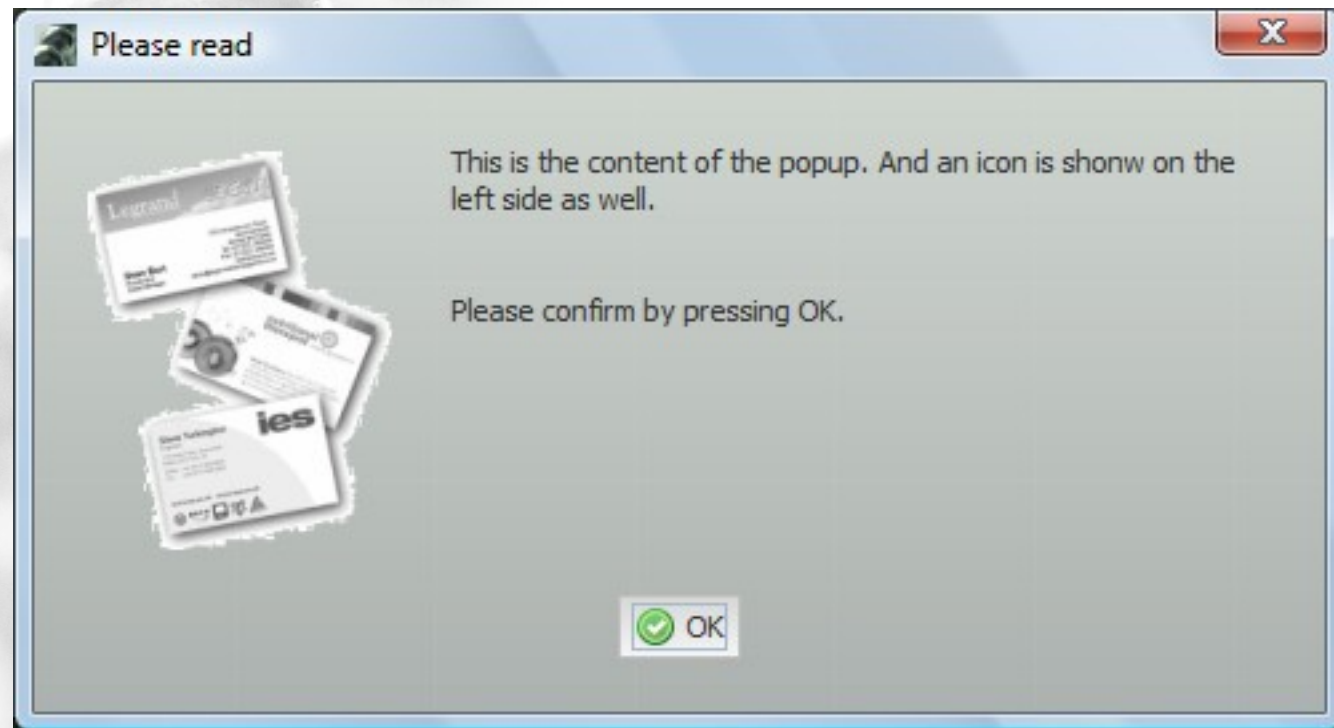
- Grids
  - Alt left, Alt right, Alt up, Alt down
  - Tab / Shift Tab
  - Pg Down/ Pg Up
  - Enter „doubleclick“, Space „click“
- Trees, in addition
  - + / -
  - Left, Right

# Modeless Popup - Extended Attributes

- `setUndecorated(true/false)`
  - ohne/mit Rahmen
- `setStartfromrootwindow(true/false)`
  - Selection of background page
  - Default: true
  - Important when starting popups from modal dialogs
- `setCloseonclickoutside(true/false)`
  - Modeless Popup disappears when clicking outside the popup

## Popup - Icon...

- OKPOpup/YESNOPopup.setIcon(...)







# Popups - With/without Workplace

- Without workplace
  - `ModalPopup.createInstance()`
  - `ModelessPopup.createInstance()`
- With workplace
  - `getOwningDispatcher().createModalPopup()`
  - `getOwningDispatcher().createModelessPopup()`

## Performance - StackedValueBinding

- Caching of values during render-response phase
- Applied for all „clean“ expressions
  - Clean: `#{d.harry.jo}`
  - Dirty: `#{d.harry.jo == true}`
- Side Effect
  - Default bindings were sometimes not properly removed from stack

## Performance - StackedValueBinding

- Caching of values during render-response phase
- Applied for all „clean“ expressions
  - Clean: `#{d.harry.jo}`
  - Dirty: `#{d.harry.jo == true}`

- `S`
  - `#{d.XXXUI.grid.rows[0].orderId.value}`
  - `#{d.XXXUI.grid.rows[0].orderId.enabled}`
  - `#{d.XXXUI.grid.rows[0].orderId.bgpaint}`
  - `#{d.XXXUI.grid.rows[0].customer.value}`
  - `#{d.XXXUI.grid.rows[0].customer.enabled}`
  - `#{d.XXXUI.grid.rows[0].customer.bgpaint}`

# Performance - IAcceleratedPropertyAccess

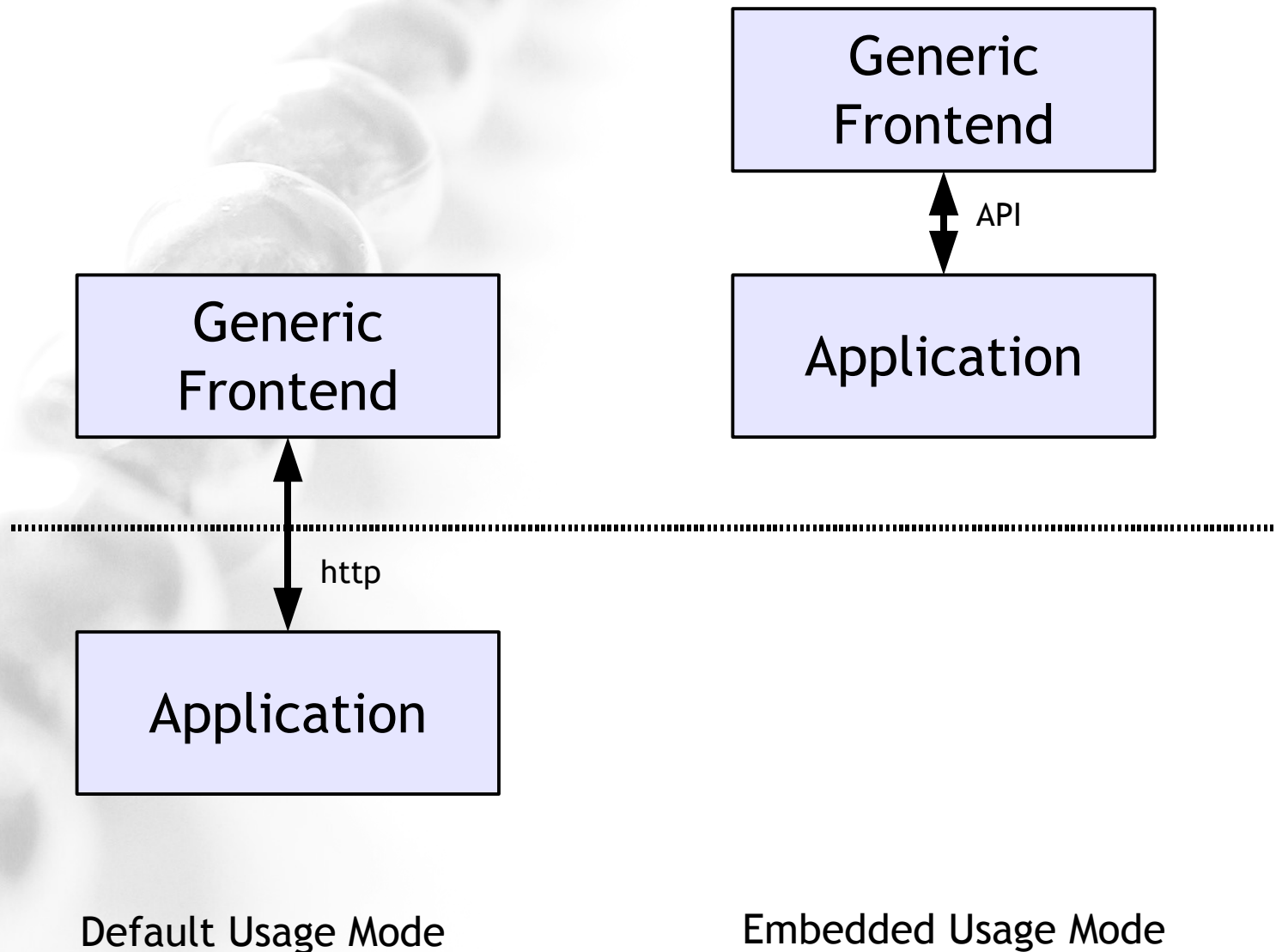
```
package org.ecInt.jsfserver.util;

/**
 * This interface is checked when reading a value of an
 * object via expression. If an object that is introspected
 * supportes this interface then it may directly pass back the
 * value. In this case no further introspection is done.
 * <br><br>
 * Background: Access via introspection is slower than un-introspected
 * access.
 * <br><br>
 * If the object cannot resolve the value then the NOT_AVAILABLE object
 * needs to be passed back, so that normal property resolution is executed.
 */
public interface IAcceleratedPropertyAccess
{
    public final static Object NOT_AVAILABLE = new Object();
    public Object getPropertyValue(String property);
}
```

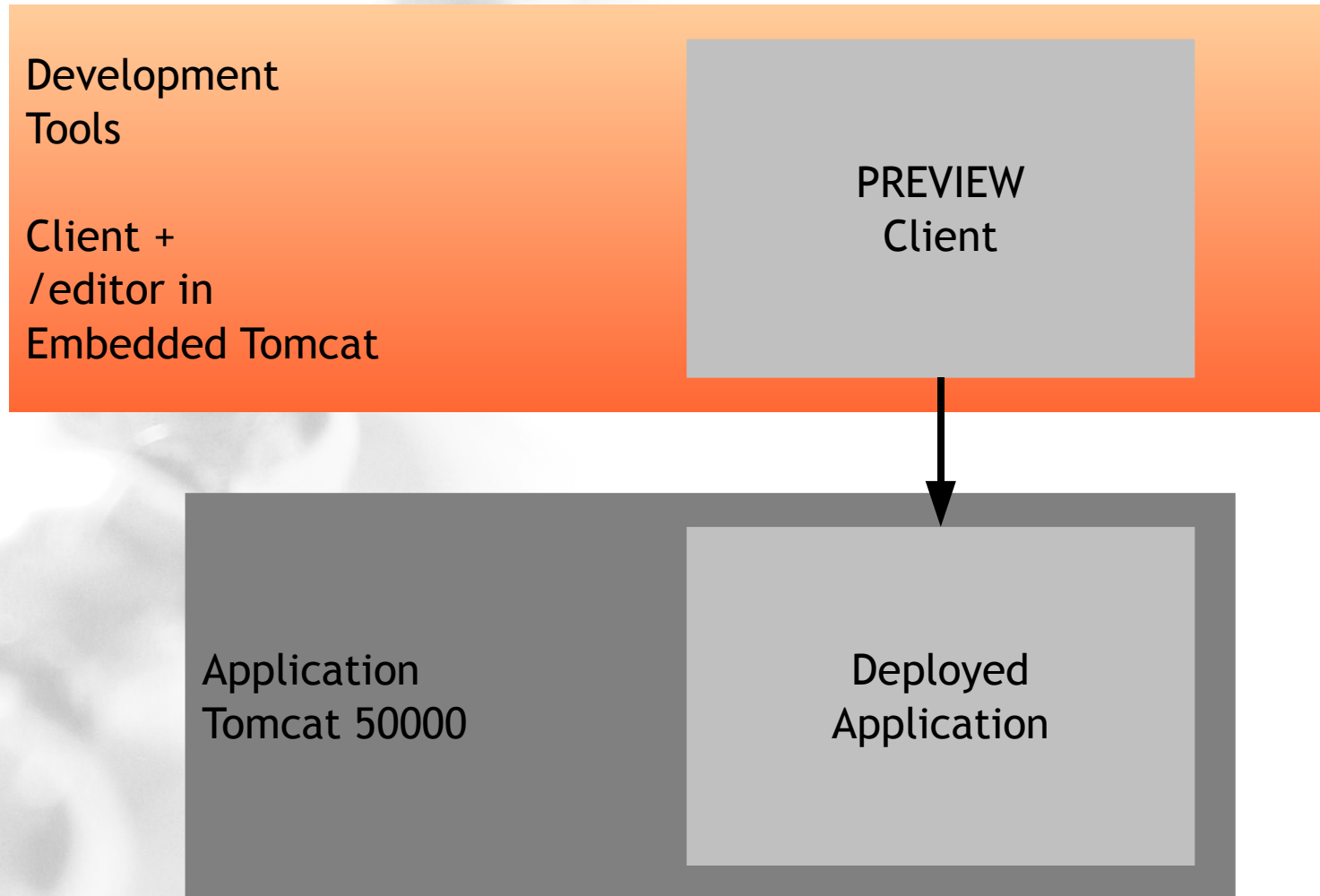
# Performance - Isolated Event Areas

- ROWINCLUDE-UPDATEONINNEREVENTONLY
  - Alle Events, die nicht innerhalb des ROWINCLUDEs verursacht wurden, bewirken keinen Update der im ROWINCLUDE referenzierten Seite
- WICHTIG für Multi-Screen-Szenarien!
- In order to override, call:
  - ThreadData.  
    getInstance().  
    registerChangeUpdatingAllAreas()

# Stand Alone / Embedded Mode (I)



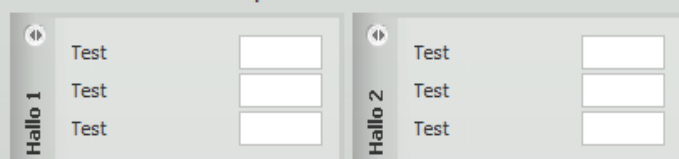
# Stand Alone / Embedded Mode (II)



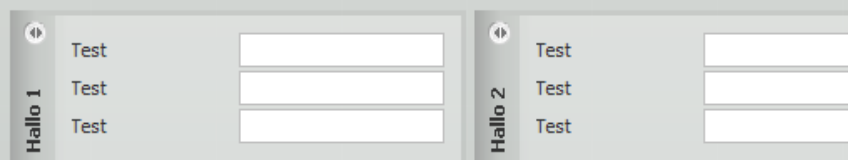
# „Vertical“ Components

- LABELVERTICAL
- FOLDABLEVERTICALPANE

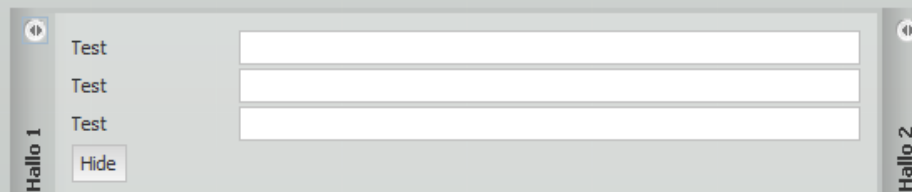
Both defined as 200 px



Both defined as 100%



Controlled From Server Side



## Vertical Label

Demo | Layout XML | Server side Java Code

CaptainCasa **Bob the Builder**

CaptainCasa  
**Bob the Builder**



# Spinner

Value between 0..10

5

+

-

Value between 0..50

20

+

-

# HTTP Header Parameters

POST /demos/faces/workplace/workplace.jsp;jsessionid=B1D8F2BC90A2861320533480F5333137 HTTP/1.1

Accept-Encoding: gzip

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ecInt-language: de

ecInt-country: DE

ecInt-orientation: ltr

ecInt-font: Tahoma

ecInt-timezone: Europe/Berlin

ecInt-client: application

ecInt-ip: 192.168.178.21

ecInt-id: ID\_72563\_34273427326

Cache-Control: no-cache

```
m_currentClientId = HttpSessionAccess.getCurrentRequest().getHeader("ecInt-id");  
m_currentClientIp = HttpSessionAccess.getCurrentRequest().getHeader("ecInt-ip");  
m_currentClientType = HttpSessionAccess.getCurrentRequest().getHeader("ecInt-client");
```

Host: localhost:50000

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

Content-Length: 180

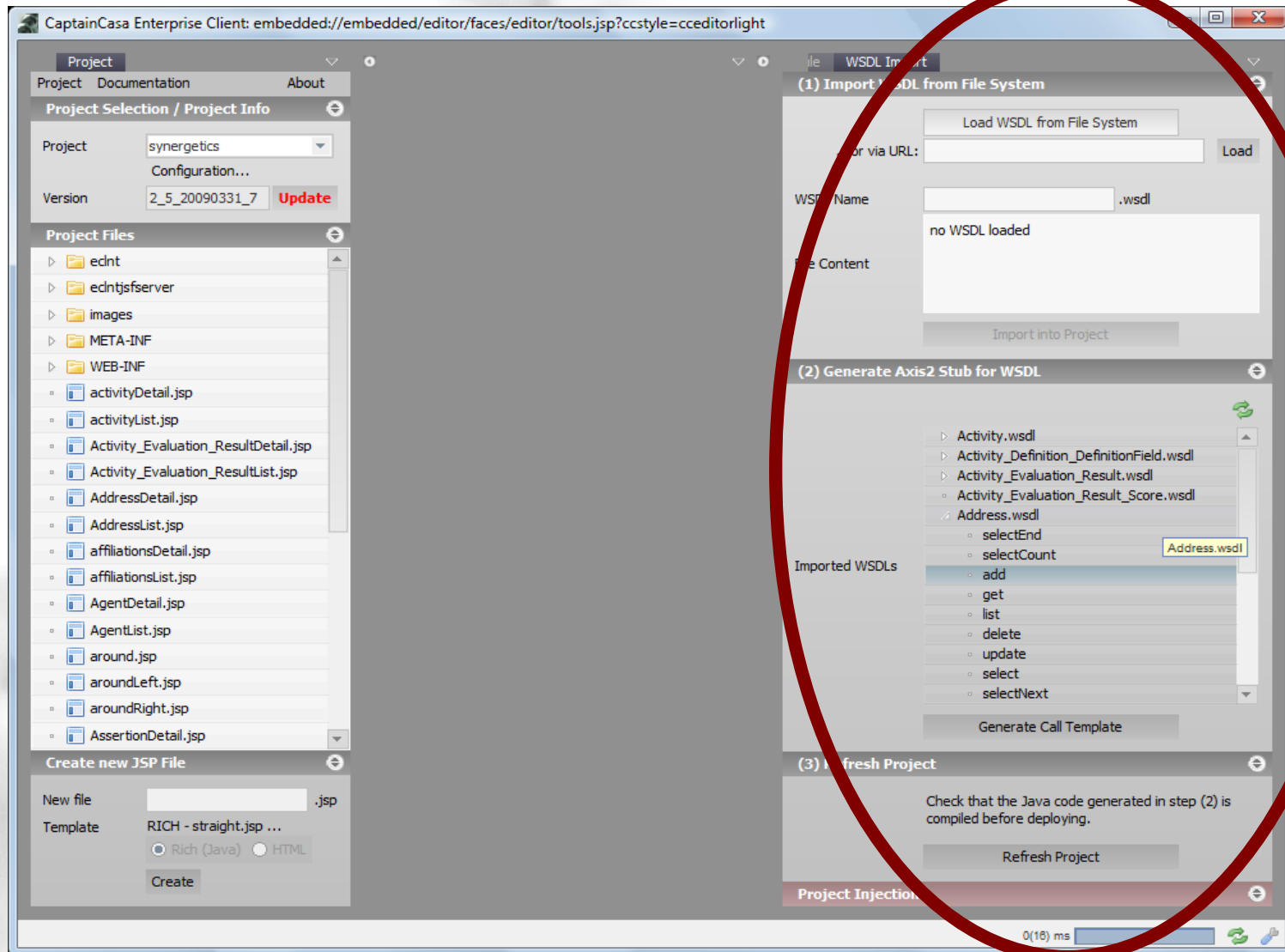
# Tools - Improved Preview Options

The screenshot displays the CaptainCasa Enterprise Client interface for developing web applications. The main window shows a project titled "workplace/demobasiccontrols.jsp" with a menu bar (File, Edit, Tools) and buttons for Save, Refresh Preview, and Preview... The interface is divided into several panels:

- Components Panel:** A tree view on the left showing a hierarchy of components. The selected component is a "t:checkbox" with a width of 120. Below the tree is a table of attributes and values.
- Preview Panel:** The central area showing a live preview of the web page. It includes a "Demo" tab, "Layout XML", and "Server side Java Code" options. The preview shows a form with a text field, a checked checkbox, radio buttons, a text area, a static combo box, and a "Summarize" button. A yellow highlight is visible on the right side of the preview.
- Style & Macro Panel:** A panel on the right side of the preview window. It includes a "Preview..." title, an "Explicit Refresh only" checkbox, a "Style" dropdown menu set to "default", and a "Show Rows" checkbox. Below these are several color swatches and a "Default Background" button.
- Messages Panel:** A panel at the bottom right showing a table with columns for "Message", "Element", and "Id".

Attribute	Value
text	Checkbox
width	120

# Tool - Extension Concept



# Applet Tag - 1.6 >= 10 „Contribution“

```
<html>

<body leftmargin=0 rightmargin=0 topmargin=0 bottommargin=0 style="overflow: hidden" onhelp="ccDoNothing();">

<script type="text/javascript">
// this code is used by APPLETCONTEXTSHOWDOCUMENT - if this is configured to
// use java script instead of directly contacting the context
function ccOpenPageInTarget (ccTarget, ccURL)
{
    var ccWindow = window.open(ccURL, ccTarget);
    ccWindow.focus();
}
function ccDoNothing()
{
    alert("ccDoNothing");
    event.returnValue = false;
}
</script>

<applet code="org.eclnt.client.page.PageApplet.class"
        archive="eclnt/lib/eclnt.jar" width="100%" height="100%"
        MAYSCRIPT">
    <param name='page' value='faces/workplace/workplace.jsp'>
    <param name="image" value="eclnt/images/splash.png">
    <param name="centerimage" value="true">
        <param name="java_arguments" value="-Xmx256m">
        <param name="java_version" value="1.6+ ">
        <param name="separate_jvm" value="true">
    <param name="MAYSCRIPT" value="true">

    <!-- This text appears in case the applet does not show up. -->
    The applet could not be loaded - please install the Java runtime
    plugin: <a href="http://www.java.com/de/download">http://www.java.com/download</a>

</applet>

</body>
</html>
```

# Lifecycle - SESSIONCLOSER Component

```
<f:view>
<h:form>
<f:subview id="workplace_workplaceg_sv">
<t:beanprocessing id="g_1" >
<t:sessioncloser id="g_2" />
</t:beanprocessing>
<t:row id="g_3" >
...
...
```

...to be placed into  
the outmost page...

# Lifecycle - Notification on Session Close

```
package org.ecInt.jsfserver.util;

public interface IHttpSessionClosedReactor
{
    /**
     * Notification about a session being destroyed (e.g. by time out management
     * of servlet container, or by administrator). You may add your code
     * to tidy up resourced that you bound within an http session.
     */
    public void reactOnClosed();
}
```

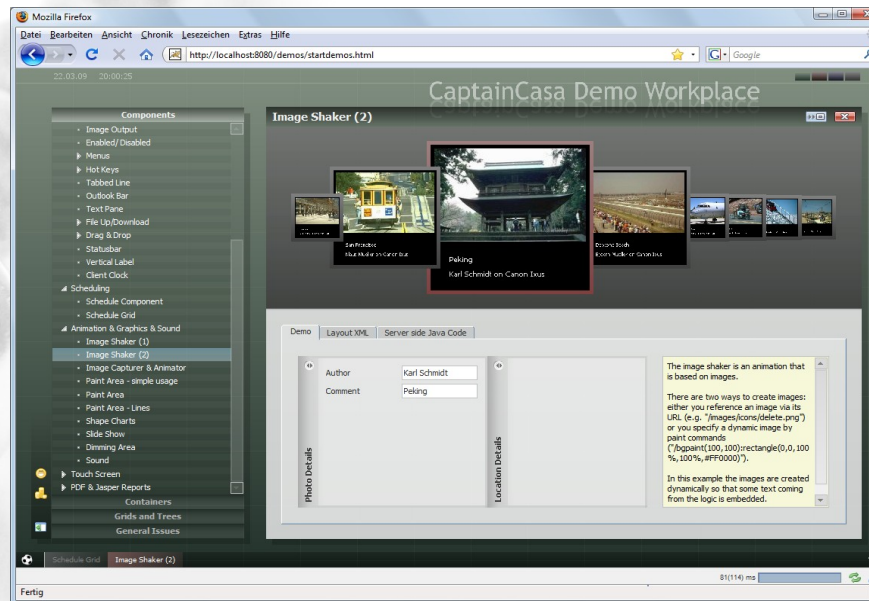
```
HttpSessionAccess.getHttpSessionListenerDelegator().add(...)
HttpSessionAccess.getHttpSessionListenerDelegator().remove(...)
```



# Client Side Integration Capabilities



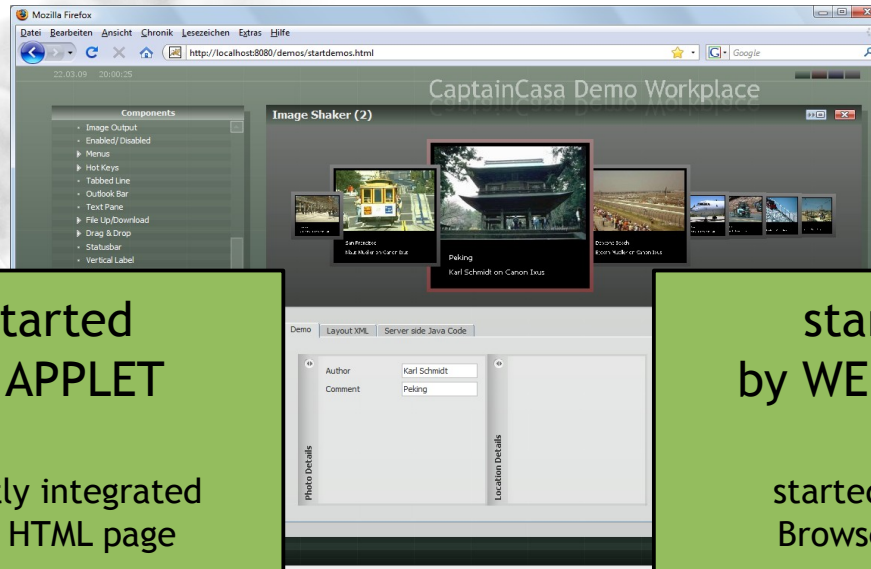
# Thin, rich Client Architecture



http(s)

JSF based Server Integration

# Thin, rich Client Architecture



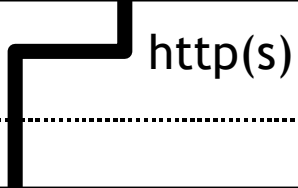
started  
as APPLET

directly integrated  
into HTML page

started  
by WEBSTART

started out of  
Browser Page

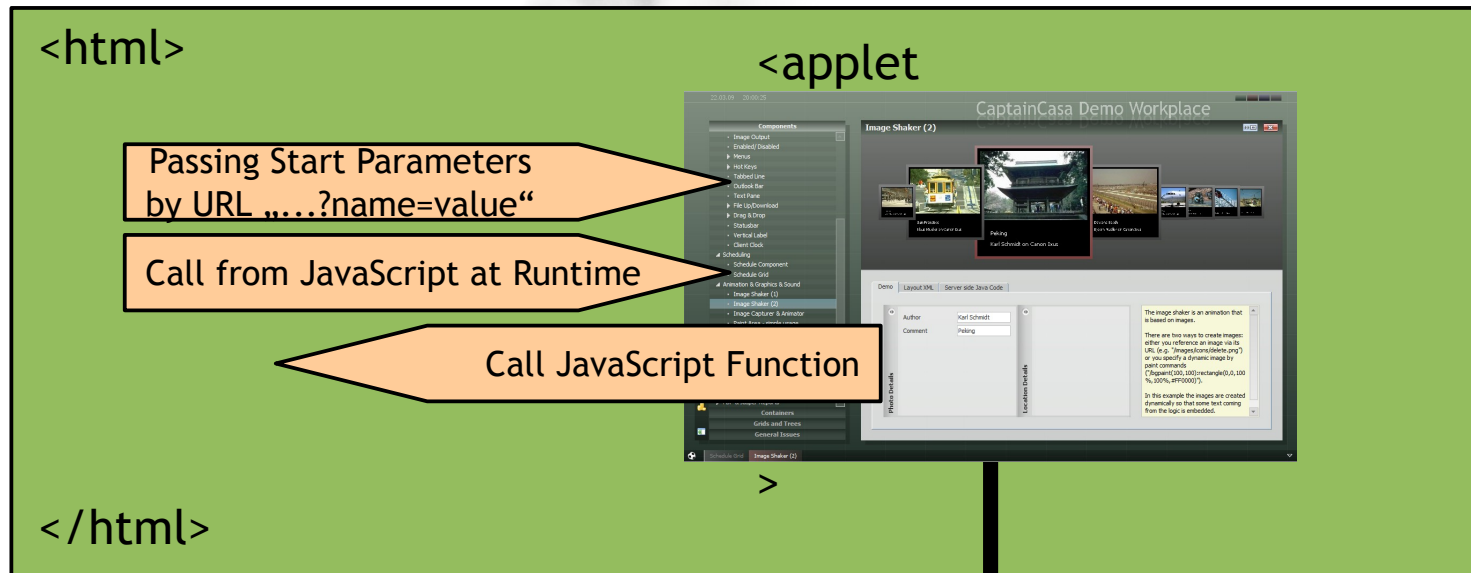
...browser's communication settings are used for http(s) communication



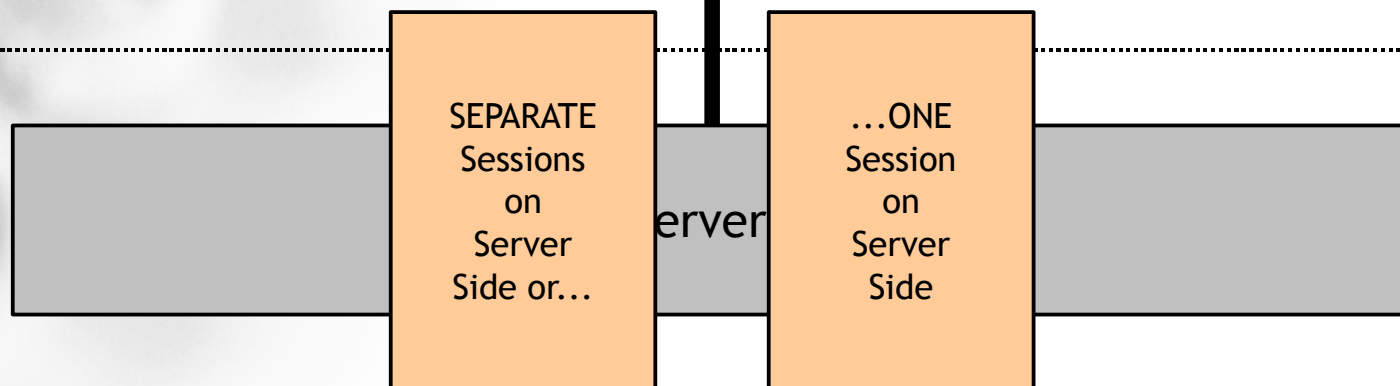
http(s)

JSF based Server Integration

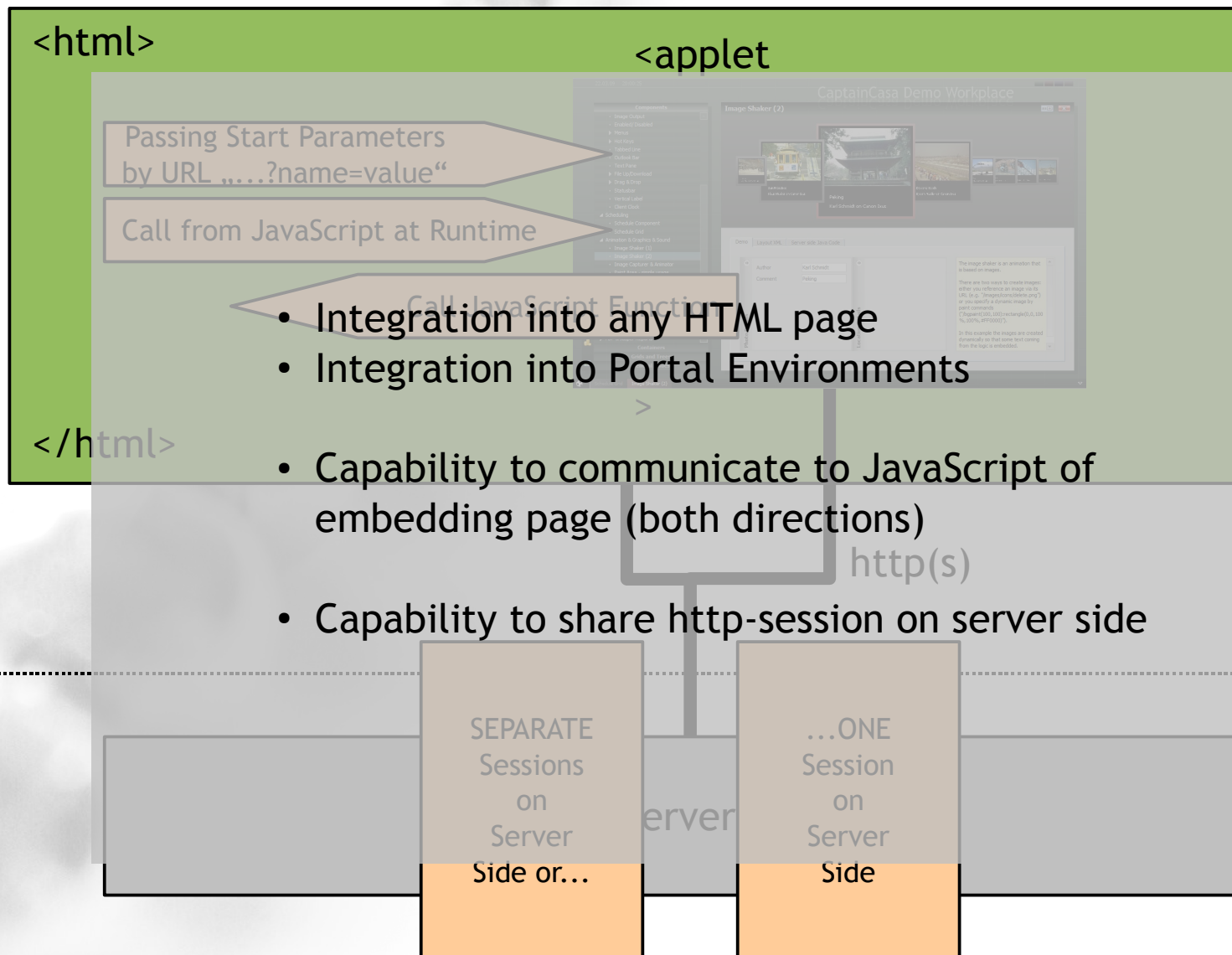
# Applet in HTML Integration



http(s)



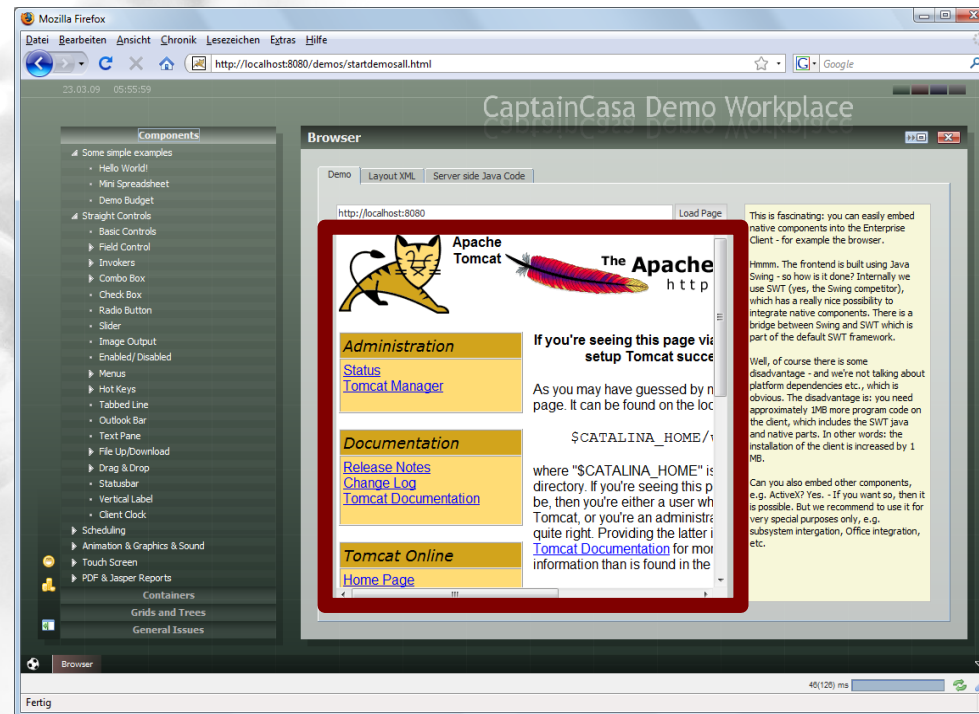
# Applet in HTML Integration



- Integration into any HTML page
- Integration into Portal Environments
- Capability to communicate to JavaScript of embedding page (both directions)
- Capability to share http-session on server side

# HTML in Enterprise Client Integration

Browser component allows native browser to be integrated into page



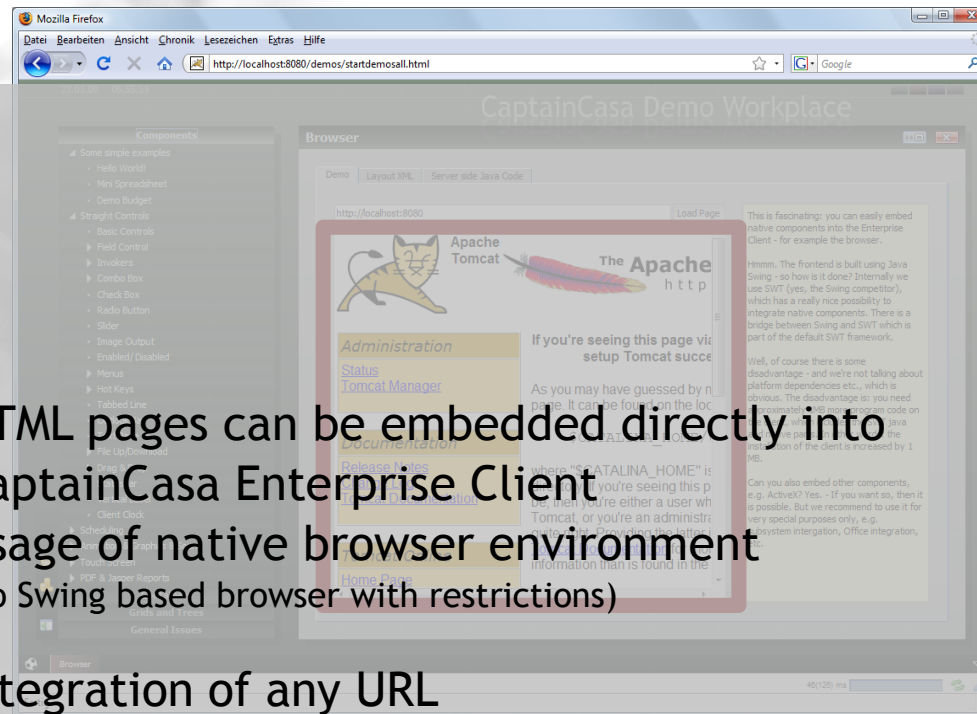
http(s)

JSF based Server Integration

# HTML in Enterprise Client Integration

Browser component allows native browser to be integrated into page

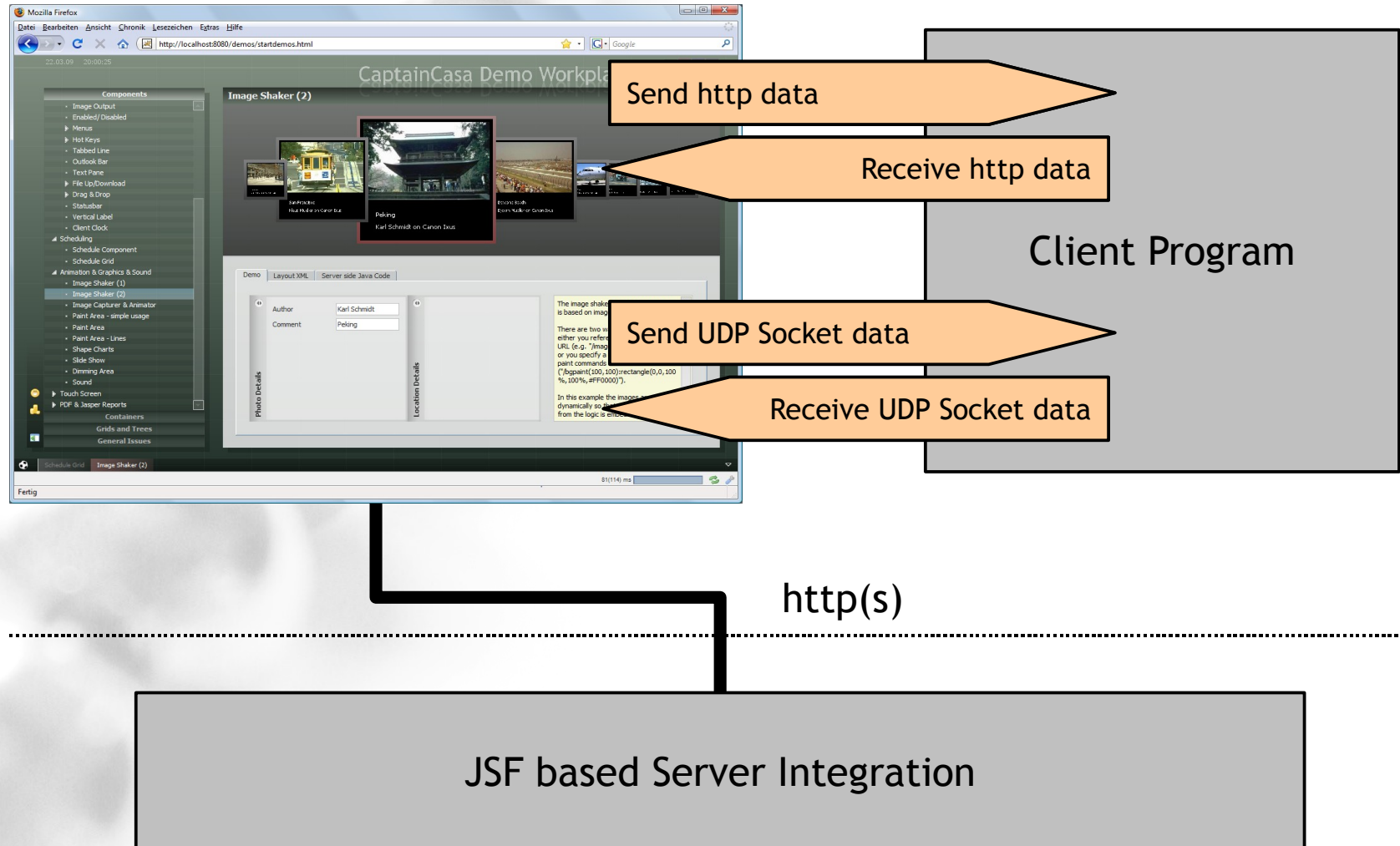
- HTML pages can be embedded directly into CaptainCasa Enterprise Client
- Usage of native browser environment (no Swing based browser with restrictions)
- Integration of any URL



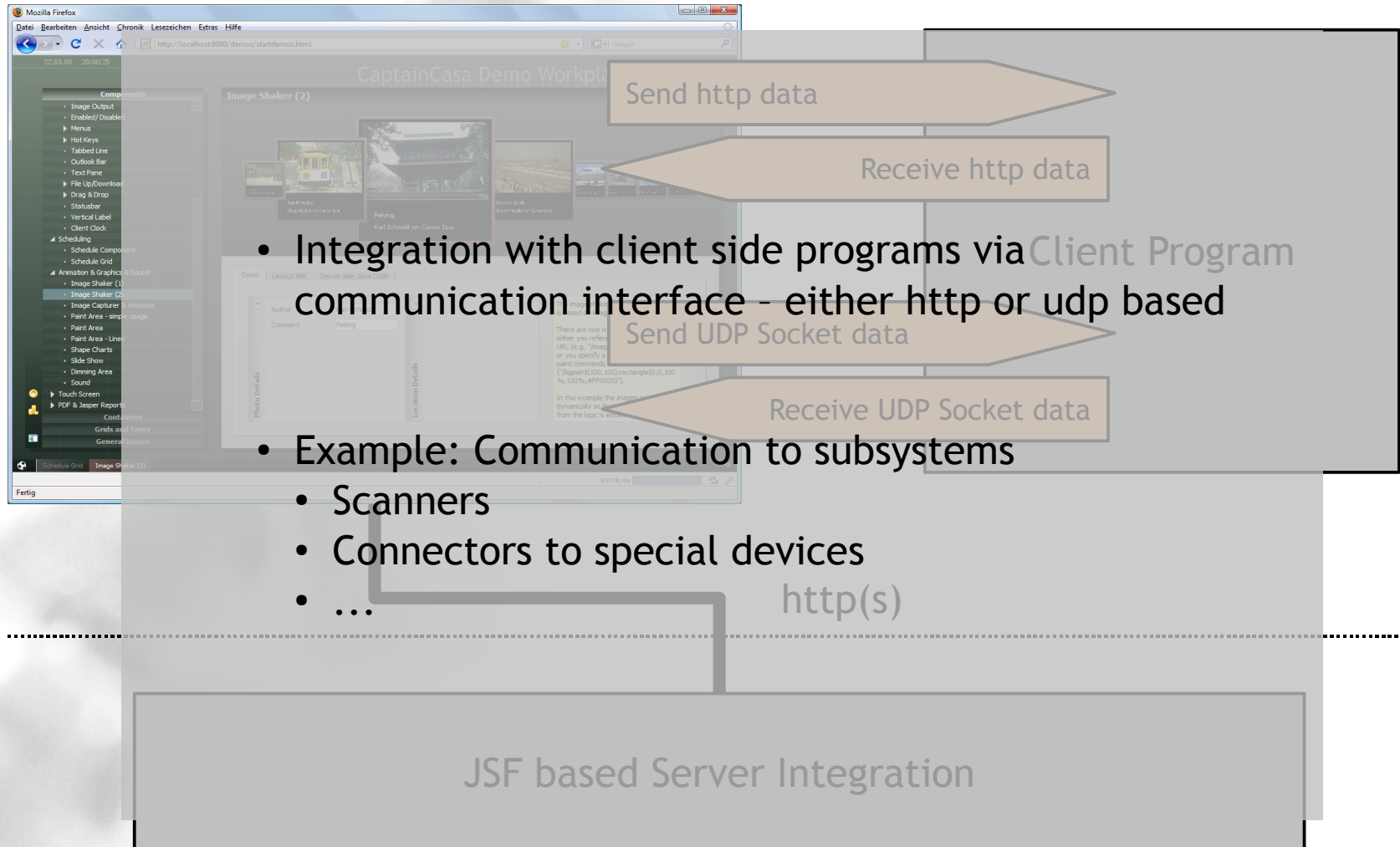
http(s)

JSF based Server Integration

# Client Side Communication Capabilities



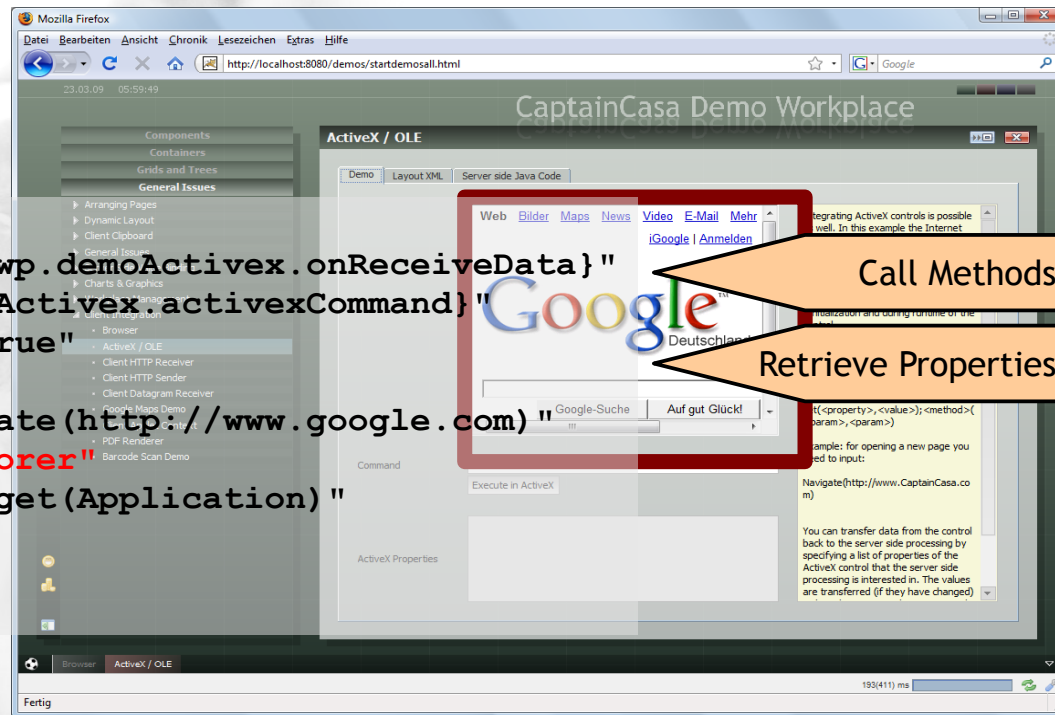
# Client Side Communication Capabilities





# ActiveX Integration (MS Windows)

```
<t:activex id="g_4"
  actionListener="#{wp.demoActivex.onReceiveData}"
  command="#{wp.demoActivex.activexCommand}"
  commandcallback="true"
  height="100%"
  initcommand="Navigate(http://www.google.com)"
  progid="Shell.Explorer"
  propertyinterest="get(Application)"
  startdelay="250"
  width="100%"/>
```



Call Methods at Runtime

Retrieve Properties at Runtime

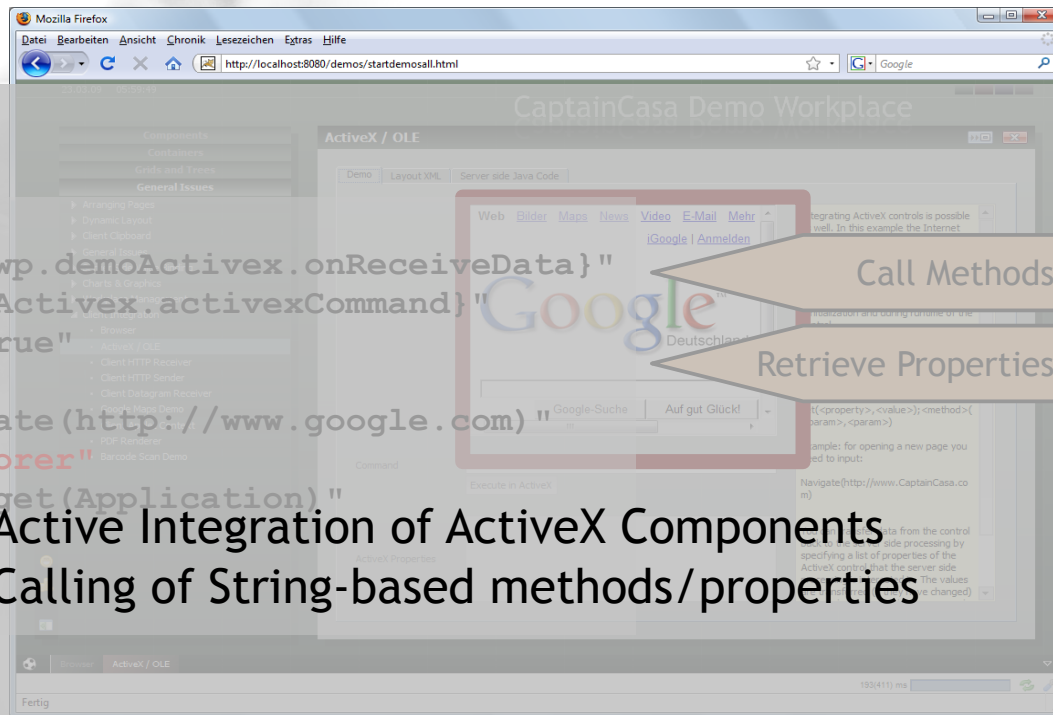
http(s)

JSF based Server Integration

# ActiveX Integration (MS Windows)

```
<t:activex id="g_4"
  actionListener="#{wp.demoActivex.onReceiveData}"
  command="#{wp.demoActivex.activexCommand}"
  commandcallback="true"
  height="100%"
  initcommand="Navigate (http://www.google.com)"
  progid="Shell.Explorer"
  propertyinterest="get (Application)"
  startdelay="250"
  width="100%"/>
```

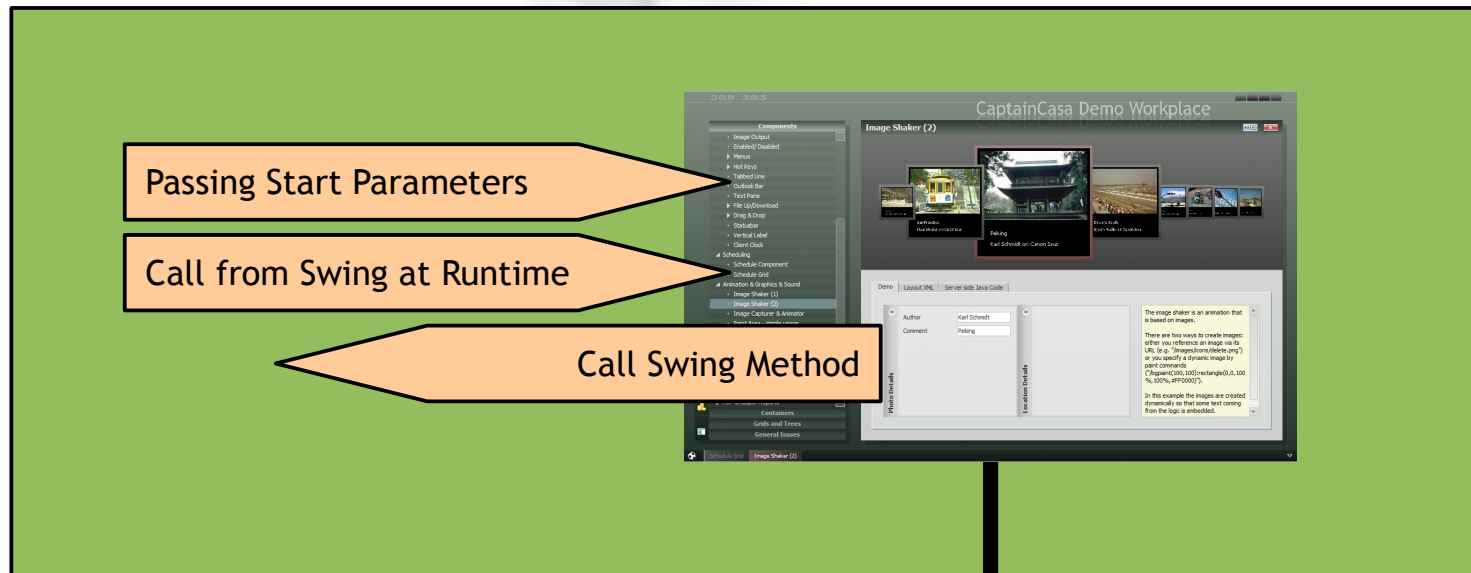
- Active Integration of ActiveX Components
- Calling of String-based methods/properties



http(s)

JSF based Server Integration

# Integration in Swing Application



http(s)

JSF based Server Integration

# Community Entwicklung



# Forum

Enterprise Client Community - Mozilla Firefox

http://www.casabacdemo.com/forum/forums/list.page

Meistbesuchte Seiten CaptainCasa Forum

## CaptainCasa Enterprise Client Community

Search Recent Topics Member Listing Back to home page  
[Register](#) / [Login](#)

The time now is: 02/04/2009 10:11:36

Forum Index [XML](#)

Forums	Topics	Messages	Last Message
<b>Category Test</b>			
<b>Development</b> All development related issues: how to deal with components, how to bind components to server side processing, etc. etc.	412	1522	01/04/2009 14:08:37 CaptainCasa →
<b>Deployment</b> All issues around deploying Enterprise Client on various client and server platforms.	22	57	01/04/2009 21:25:56 mreich →
<b>Product</b> All non-short term technical questions about the product. Licensing, future steps, etc. Moderators: Moderators	76	103	30/03/2009 12:51:26 CaptainCasa →
<b>Community</b> CaptainCasa Enterprise Client is used and managed by a community of companies. Discussions / Questions about this community can be placed into this forum.	21	35	23/03/2009 10:40:23 CaptainCasa →

**Who is online**

Our users have posted a total of 1,717 messages  
 We have 61 registered users  
 The newest registered user is tdrasch

There are 1 online users: 0 registered, 1 guest(s) [ Administrator ] [ Moderator ]  
 Most users ever online was 11 on 04/06/2008 16:26:38  
 Connected users: Anonymous

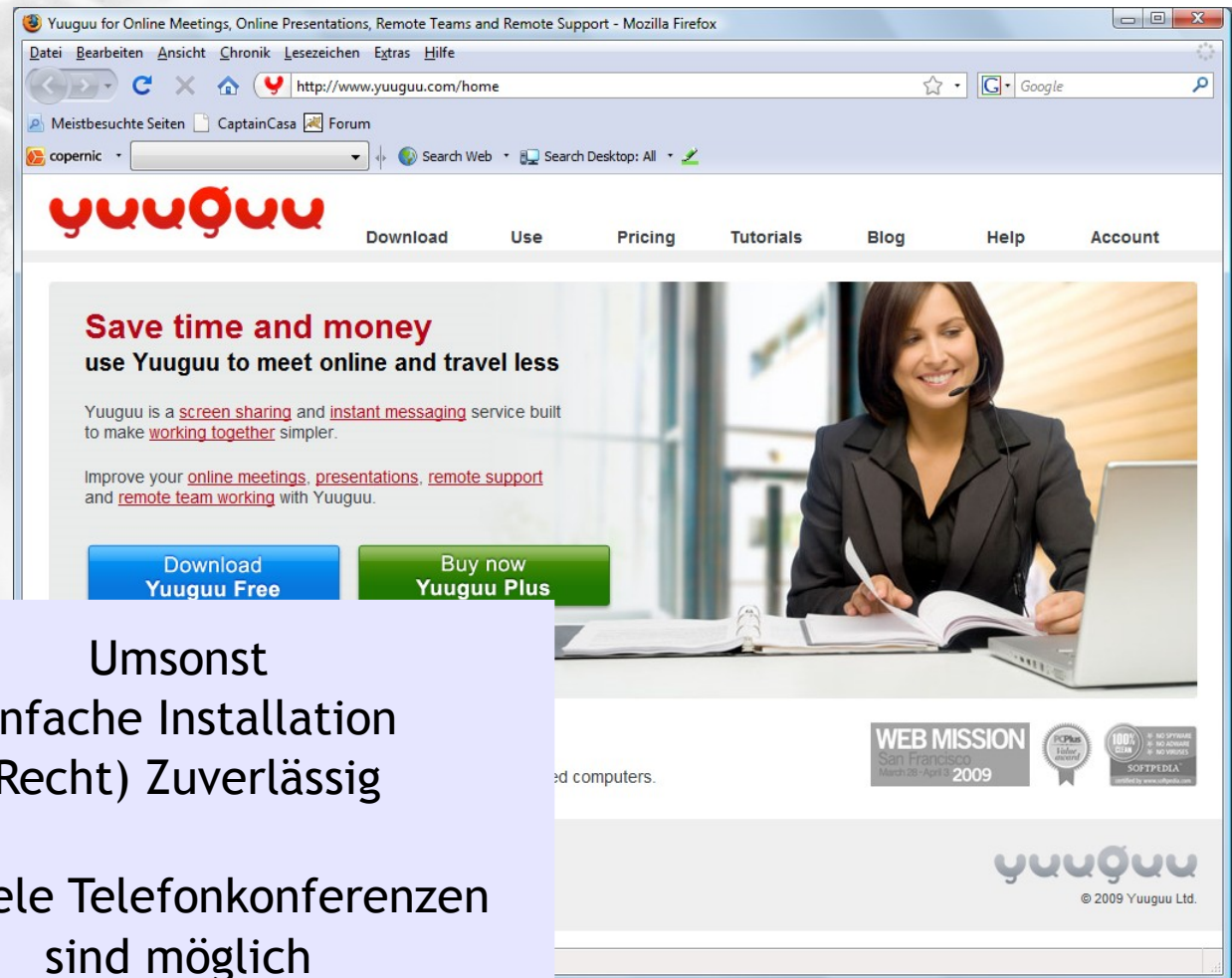
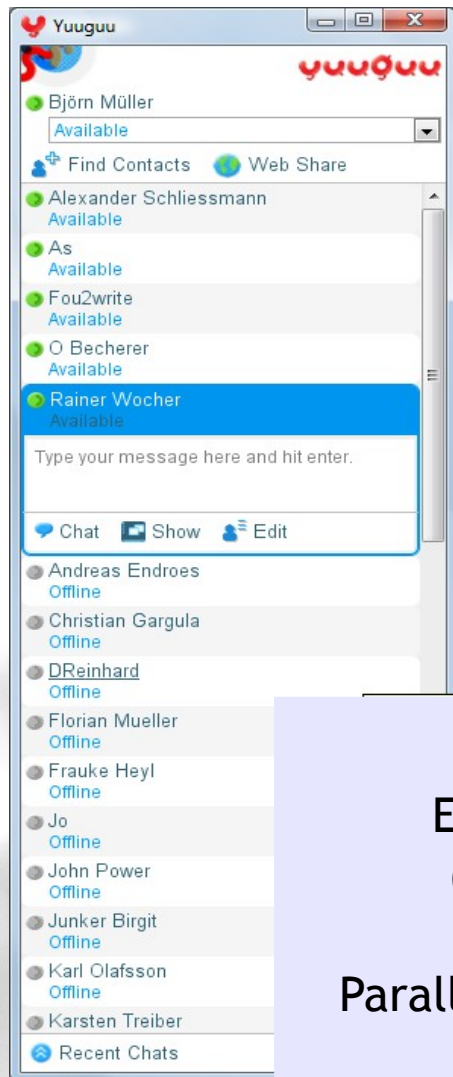
**Login**

Username:  Password:  Log me on automatically each visit

New Messages 
 No new messages 
 Blocked Forum

Fertig

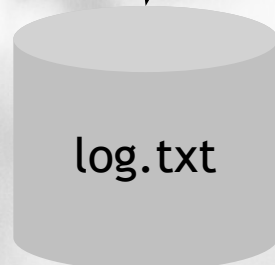
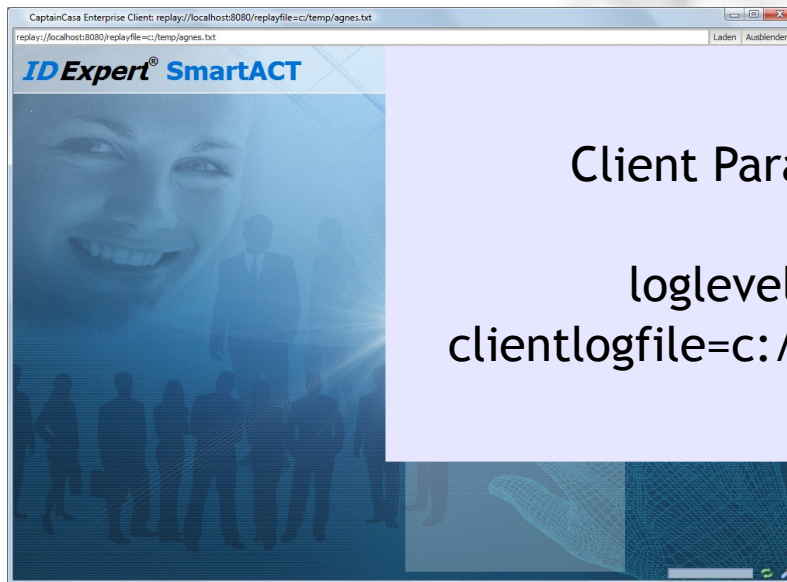
# Yuuguu - Screensharing



Umsonst  
Einfache Installation  
(Recht) Zuverlässig

Parallele Telefonkonferenzen  
sind möglich

# Client Log File Versand



# Schulungen

Seminare Enterprise Client - Mozilla Firefox

http://www.versatec.de/Seminare-Enterprise-Client.htm

## VersaTec IT Services GmbH

### Seminare Enterprise Client

- Home
- Service-Angebote
- Software-Produkte
- Seminare
  - ITIL
  - Java\_OO\_UML
  - JavaServer Faces
  - Java Enterprise Client
    - XML\_DOM\_SAX
    - SOA Gateway
    - SAP R/3
    - MS Access\_Excel\_VBA
    - Unix
    - Softskills
    - Seminar-Termine
    - Trainer
    - Personalberatung
    - Jobs
    - Kunden\_Referenzen
    - Partner
    - Kontakt\_Impressum

### Themen

Der CaptainCasa Enterprise Client ist ein communitybasiertes, praxiserprobtes Framework zur effizienten Gestaltung hochwertiger Benutzeroberflächen. Die Oberflächen laufen innerhalb einer Browserumgebung vor der serverseitigen, auf JSF (JavaServer Faces) basierten Anwendung. Weitere Informationen unter <http://www.captaincasa.com>.

Rich-Client Architekturen unterstützen insbesondere Anwendungen mit hohen Anforderungen an Interaktion, Dateneingaben oder grafischen Darstellungen sowie schnelle Ausführungen und Kommunikation mit dem Backend. Im Vergleich zu reinen Browser-Lösungen muss auf dem Arbeitsplatz eine JVM zur Verfügung stehen - das stellt aber in den meisten Fällen keine Einschränkung dar.

- [Rich Client Entwicklung mit CaptainCasa ... in a Nutshell](#)
- [Rich Client Entwicklung - Grundkurs](#)
- [Rich Client Entwicklung - Anwendungsintegration](#)

### Rich Client Entwicklung mit CaptainCasa ... in a Nutshell

#### Overview

Dauer:  1 Tag

Ziel:  Binnen einen Tages erhält der Teilnehmer eine "Druckbetankung", die zeigt, welche Möglichkeiten die CaptainCasa Rich-Client-Umgebung bietet und wie sie vor einer

Fertig



# Services

- Versatec, Mainz
- Pilz & Partner, Stuttgart, Berlin
- Object International, Stuttgart
- Fumix, Heidelberg
- USU AG, Stuttgart
- Indato, Bern

# Produkt Partnerschaften

Quality First Software GmbH & QF-Test - Das Java GUI Testtool - Mozilla Firefox

http://www.qf-test.de/

Meistbesuchte Seiten CaptainCasa Forum

**QFS**  
Quality First Software

Home Kontakt Suchen

Produkt  
Download & Evaluation  
Support  
Preise  
FAQ  
Referenzen  
Neuigkeiten & Events  
Firma  
Open Source

**QF-TEST**  
Professionelles GUI-Testen  
für Java und Web

Mit QF-Test bietet Quality First Software ein etabliertes Produkt zur Automatisierung von Tests für Java- und Web-Anwendungen mit grafischer Benutzeroberfläche an. Weit über die unverzichtbaren Unittests hinaus deckt QF-Test viele verschiedene Arten von Fehlern im System aus Anwendersicht auf.

Mit QF-Test automatisierte Tests tragen in hohem Maß dazu bei, Zeitaufwand und Kosten der Softwarequalitätssicherung zu reduzieren. Reproduzierbare Zuverlässigkeit sorgt für höhere Qualität und macht oft den Unterschied zwischen früherem Markteintritt und Verzögerungen im Zeitplan.

**QF-TEST auf einen Blick**

- Testautomatisierung von Swing und Eclipse/SWT/RCP und Web
- Plattformübergreifende System- und Lasttests
- Benutzerfreundlich - schneller Einstieg
- Robuste Tests mit zuverlässiger Wiedererkennung
- Sehr gutes Preis-Leistungsverhältnis und hoher ROI
- Schneller und kompetenter Support direkt vom Hersteller

Demo-Video ansehen

Download einer freien Testversion

Download von QF-Test

**Letzte Neuigkeiten**

27. März 2009

**QF-Test Version 3.0.2 freigegeben**

Version 3.0.2 ist eine Wartungsversion für den aktuellen Produktzweig von QF-Test. Neben der Beseitigung einiger Bugs bringt diese Version auch neue Features mit, darunter Unterstützung für die aktuellen Eclipse/SWT Versionen 3.4.2 und 3.5M6 sowie eine Erweiterung des Daemon API. [Release notes](#), [Download](#), [Über QF-Test](#)

10. März 2009

Fertig



# Aha Erlebnisse vor der Produktivsetzung

# Memory (I)

- Wann wird wie aufgeräumt?
  - Identifizieren von relevanten Zeitpunkten
  - Profilen des erfolgreichen Aufräumens
- Profiling Tool der Wahl (neben anderen)
  - TJIMP

## Memory (II)

- Wie hoch ist die Memorybenutzung pro Benutzer?
  - Beobachtungswürdig:
    - Grids...!
- Mittel
  - ARRAYGRID Nutzung für Anzeige Grids
  - Lazy Loading in der Anwendung

# Logging

- CaptainCasa Konfiguration
  - /ecIntjsfserver/config/logging.xml
- Für Produktivgang
  - level="SEVERE"
  - Auf keinen Fall: level="INFO"
  - Auf keinen Fall: console="TRUE"

# Single Process >> Cluster

- Wie wird auf Dateien zugegriffen?
  - Gut:
    - ClassLoader
    - WebResourceReader / ServletContext
  - Schlecht:
    - Direktes File I/O
- Wo werden „static“s verwendet
  - Static <==> Session Statics
  - Session Statics in die http-session oder an zentrale managed beans (e.g. Dispatcher)

# http(s), reverse proxy, ...

- Frühzeitiges Testen unter Live-Kommunikationsbedingungen - inklusive Proxies
  - Es tauchen die wildesten Dinge auf...!



## Welche JARs in den Client?

- Nur die JARs laden, die notwendig sind!
- Insbesondere: swt.jar nicht unnötig nach vorne laden, die verhindert manchmal den Starten des Applets

# Einbindung in äußere HTML Anwendung

- Frühzeitiges Einbinden des Applets!
  - Session: CaptainCasa Applet läuft per Default in einer eigenen http Session - gibt es eventuell doch Gründe, die Session zu sharen?