HP webOS

Native Web-Applikationen

Björn Adelberg

17. November 2011

Gliederung

- Teil1: Das Betriebssystem HP webOS
- Teil2: Entwickeln unter HP webOS

Teil 1

Das Betriebssystem HP webOS

Inhaltsverzeichnis

Übersicht

Die webOS Versionen

Die webOS Geräte

WebOS Besonderheiten
- Multitasking
- Synergy
- Just type
- Exhibition Mode

HP webOS unter der Haube
- Architektur
- WebOS App Typen

Inhaltsverzeichnis

Übersicht

Die webOS Versionen

Die webOS Geräte

WebOS Besonderheiten
- Multitasking
- Synergy
- Just type
- Exhibition Mode

HP webOS unter der Haube
- Architektur
- WebOS App Typen

Inhaltsverzeichnis

- Übersicht
- Die webOS Versionen
- Die webOS Geräte
- WebOS Besonderheiten
  - Multitasking
  - Synergy
  - Just type
  - Exhibition Mode
- HP webOS unter der Haube
  - Architektur
  - WebOS App Typen

Inhaltsverzeichnis

- Übersicht
- Die webOS Versionen
- Die webOS Geräte
- WebOS Besonderheiten
  - Multitasking
  - Synergy
  - Just type
  - Exhibition Mode
- HP webOS unter der Haube
  - Architektur
  - WebOS App Typen

Inhaltsverzeichnis

- Übersicht
- Die webOS Versionen
- Die webOS Geräte
- WebOS Besonderheiten
  - Multitasking
  - Synergy
  - Just type
  - Exhibition Mode
- HP webOS unter der Haube
  - Architektur
  - WebOS App Typen

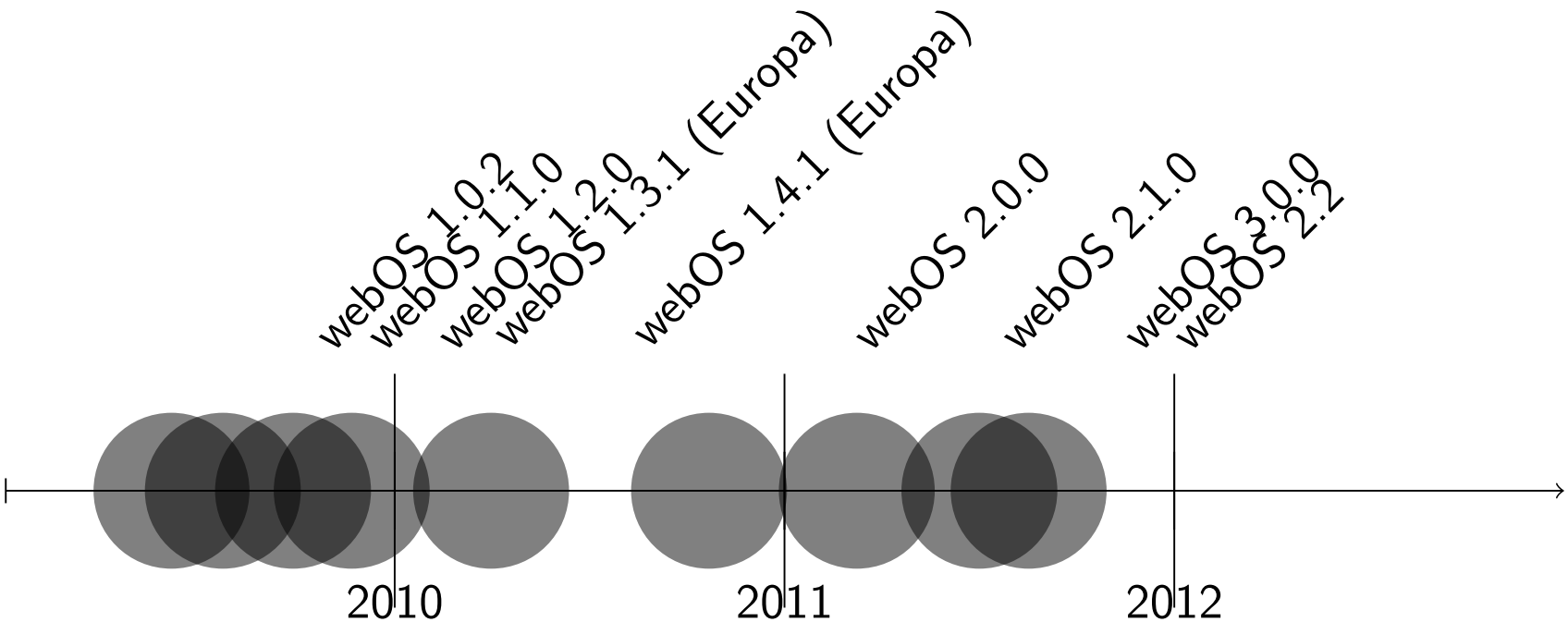
Frage

Was weißt Du über HP webOS?

Fakten

- ehemaliger Inhaber war Palm, jetzt in Besitz von HP
- erstmals auf *Consumer Electronics Show* am 8. Januar 2009 in Las Vegas vorgestellt
- Multitasking, Synergy, Exhibition Mode
- kleine aber starke Community
- ca. 7000 Apps (Quelle: precentral.net Juni 2011)
- einfacher Zugriff auf das Betriebssystem
- Linux Kernel, Luna
- Änderungen am Kernel Open Source, ansonsten Closed Source
- Apps auf Basis von JavaScript, CSS und HTML

Timeline von webOS



Geräte von Palm

Palm Pre

Palm Pixi

Palm Pre Plus

Palm Pixi Plus

Palm Pre 2

Geräte von HP

HP Veer

HP Pre 3

HP Touchpad

webOS Besonderheiten

- Multitasking
- Synergy
- Just type
- Exhibition Mode

webOS Besonderheiten

- Multitasking
- Synergy
- Just type
- Exhibition Mode

webOS Besonderheiten

- Multitasking
- Synergy
- Just type
- Exhibition Mode

webOS Besonderheiten

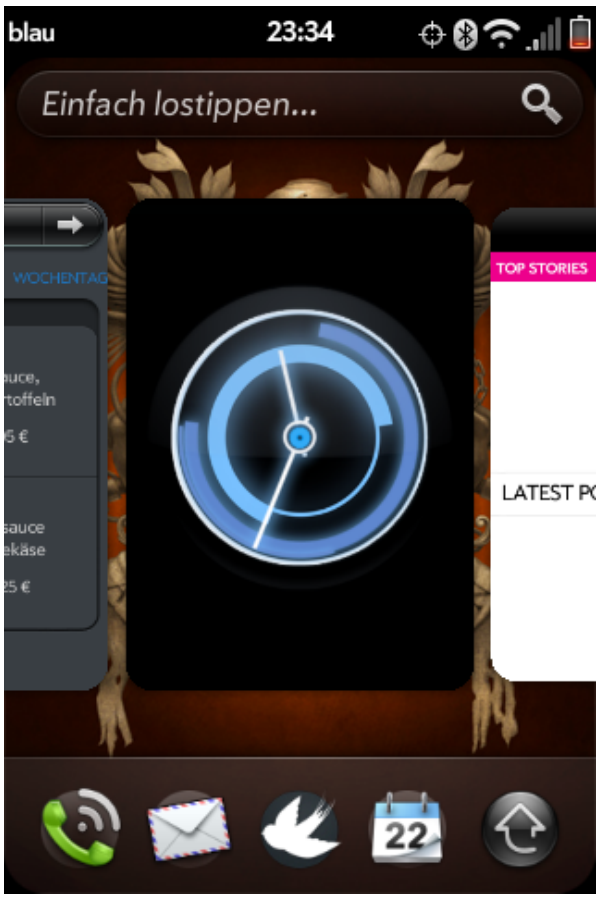
- Multitasking
- Synergy
- Just type
- Exhibition Mode

Multitasking

Multitasking

Der Begriff Multitasking (engl.) bzw. Mehrprozessbetrieb bezeichnet die Fähigkeit eines Betriebssystems, mehrere Aufgaben (Tasks) nebenläufig auszuführen. Dabei werden die verschiedenen Prozesse in so kurzen Abständen immer abwechselnd aktiviert, dass der Eindruck der Gleichzeitigkeit entsteht. (Quelle: Wikipedia)

In webOS verkörpert das Konzept der *Karten (Cards)* den Mehrprozessbetrieb.



Synergy

Synergy

Synergy vereinfacht den Umgang mit Online-Accounts indem es diese nach Einsatzzweck (E-Mail, Chat, Kalender, Dateiablage, etc) gruppiert und den jeweiligen Anwendungen zur Verfügung stellt.

Just type

Just type

Just type (Einfach lostippen...) vereinfacht die Bedienung eines webOS Gerätes enorm. Es ist möglich textbasierte Aktionen zu starten.

- Kontakte öffnen
- Suchmaschine verwenden
- Favoriten und Verlauf durchsuchen
- Eine App starten und durchsuchen
- Schnellaktionen (FB Status, Tweet, E-Mail, Notizen, Aufgaben) mit Vorgabetext ausführen; Entwickler kann seine App hier anbinden

Exhibition Mode

Exhibition Mode

Präsentations-Modus in welchen gewechselt werden kann. Wird automatisch aktiviert beim laden des Gerätes über den Touchstone.

- Wetteranzeige
- Timelines
- und vieles mehr

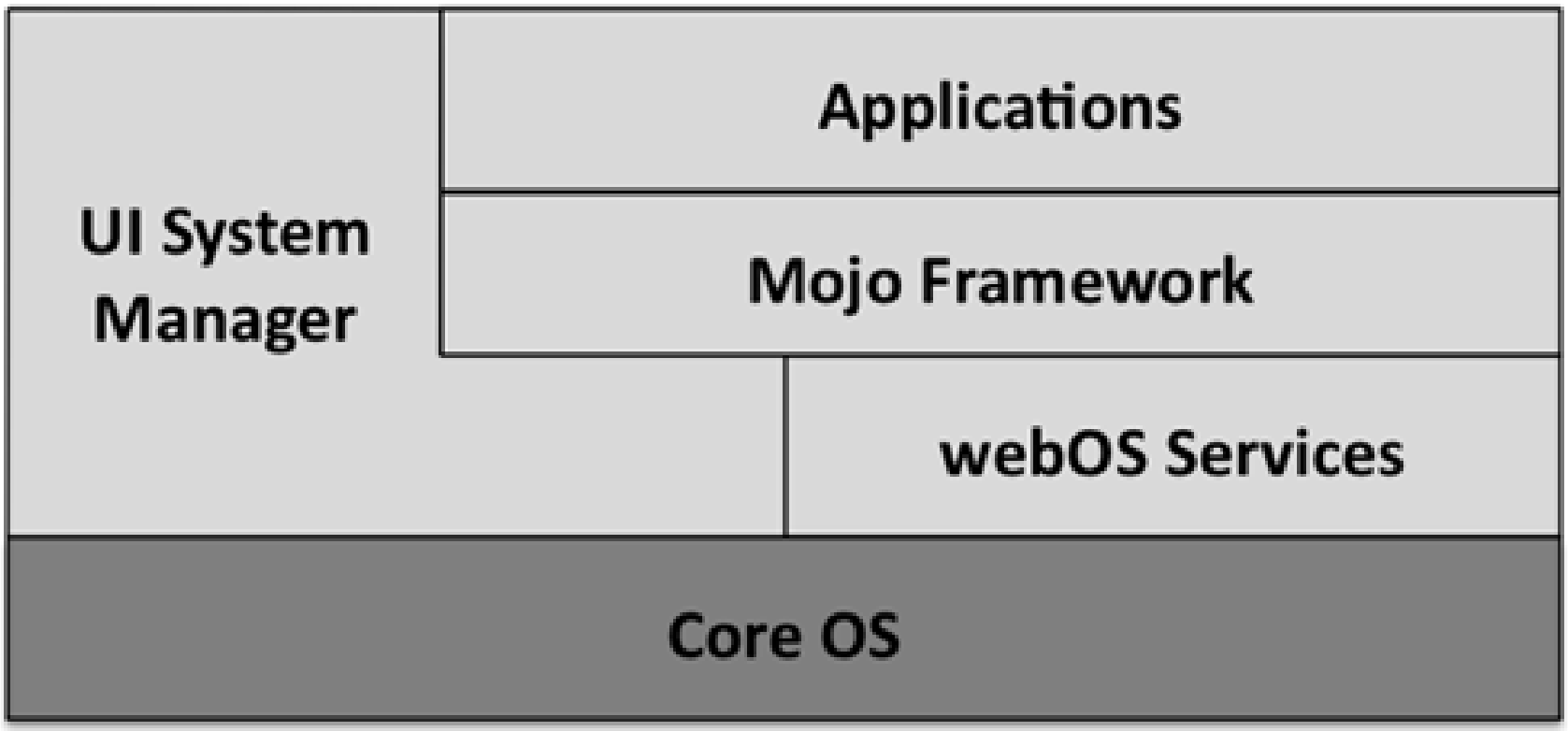
Frage

Hast Du noch Fragen?

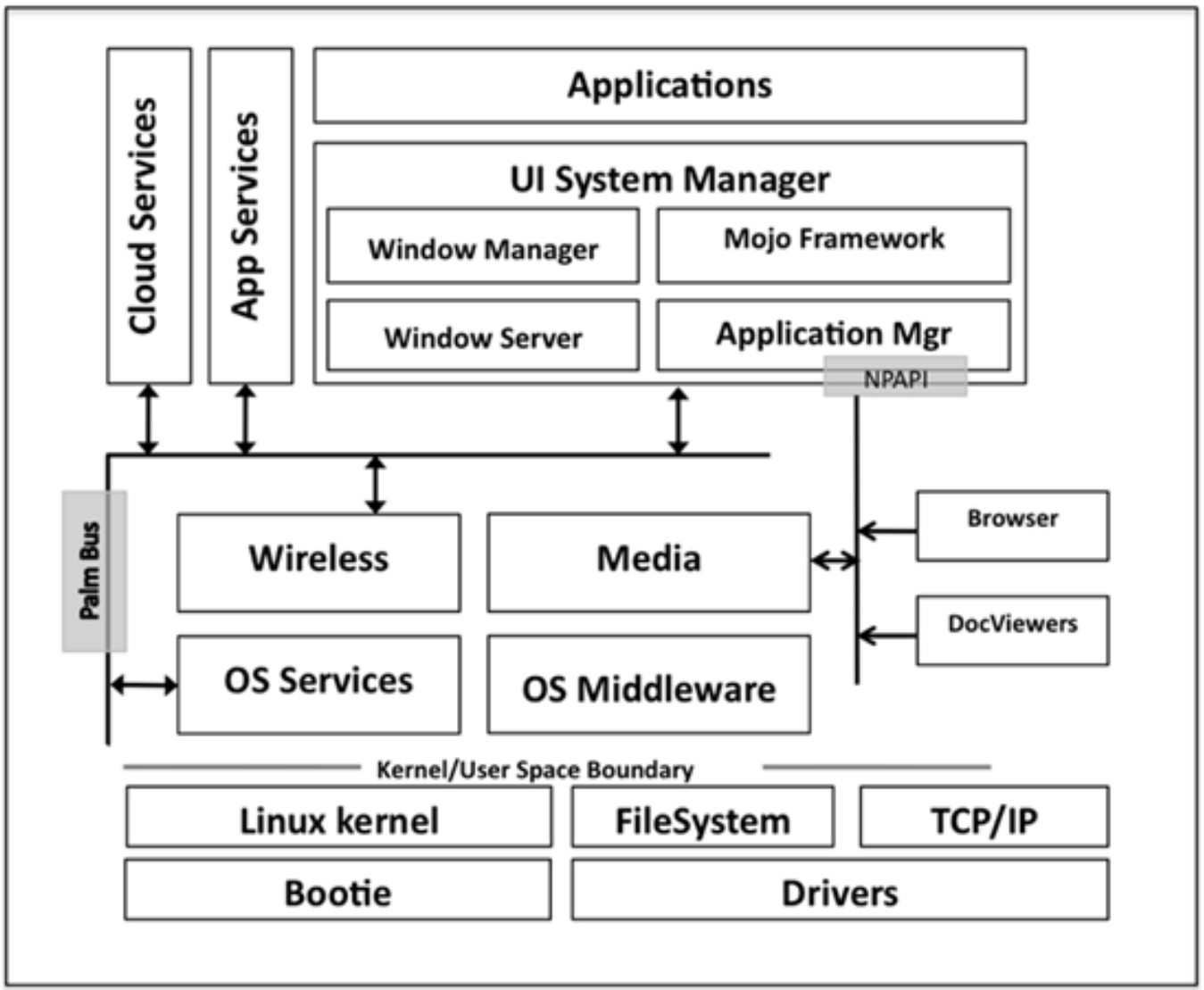
WebOS Architektur - die Fakten

- basiert auf Linux Kernel 2.6
- kein X
- Benutzer interagiert mit verschiedensten Applikation und UI System Manager
- weiterhin existieren webOS Services

Vereinfachte webOS Architektur



WebOS Architektur



top Ausgabe

```
top - 22:06:55 up 1 day, 11:38,  2 users,  load average: 8.57, 8.48, 8.32
Tasks: 146 total,   1 running, 145 sleeping,   0 stopped,   0 zombie
Cpu(s): 12.2%us, 16.9%sy,  0.0%ni, 70.6%id,  0.3%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    389244k total,   378564k used,    10680k free,     6724k buffers
Swap:   516088k total,    48000k used,   468088k free,    86976k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1475 root      19  -1  145m  54m  40m S 14.4 14.4   7:35.97 LunaSysMgr
26764 root      19  -1 65144  19m  13m S  6.6  5.2   0:01.59 Xsdl
26808 root      20   0  2660 1264  960 R  3.4  0.3   0:00.62 top
 1108 root      17  -3 84532 3180 1756 S  3.1  0.8   5:34.52 hidd
 1497 root      19  -1  366m 149m  25m S  1.3 39.4  28:28.27 WebAppMgr
  985 root      20   0  3188  972  720 S  0.3  0.2   2:49.50 pmsyslogd
 1150 root      20   0 24500 2440 1840 S  0.3  0.6  17:45.45 powerd
    1 root      20   0  2176  852  532 S  0.0  0.2   0:11.17 upstart
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.13 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.38 ksoftirqd/0
    4 root      20   0     0    0    0 S  0.0  0.0   1:36.17 events/0
    5 root      20   0     0    0    0 S  0.0  0.0   0:00.01 khelper
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
    7 root     -51   0     0    0    0 S  0.0  0.0   0:00.10 irq/155-pm8058-
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.02 sync_supers
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.15 bdi-default
   10 root      20   0     0    0    0 S  0.0  0.0   0:02.31 kblockd/0
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.09 ksuspend_usbd
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.15 khubd
   13 root      20   0     0    0    0 S  0.0  0.0   0:00.46 kmmcd
   14 root      20   0     0    0    0 S  0.0  0.0   0:00.00 bluetooth
```

WebOS App Typen

- Mojo Framework
- Enyo Framework
- PDK Apps in C/C++
- JavaScript Services (Foundations, Node.js Add-ons)
- Synergy Konnektoren
- PhoneGap

Teil 2

Entwickeln unter HP webOS

Das Framework Mojo

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Mojo

- Bauen, Installieren und Aufrufen
- Testen

Widgets

Services

Mensa App

Fragen

Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Mojo

- Bauen, Installieren und Aufrufen
- Testen

Widgets

Services

Mensa App

Fragen

Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

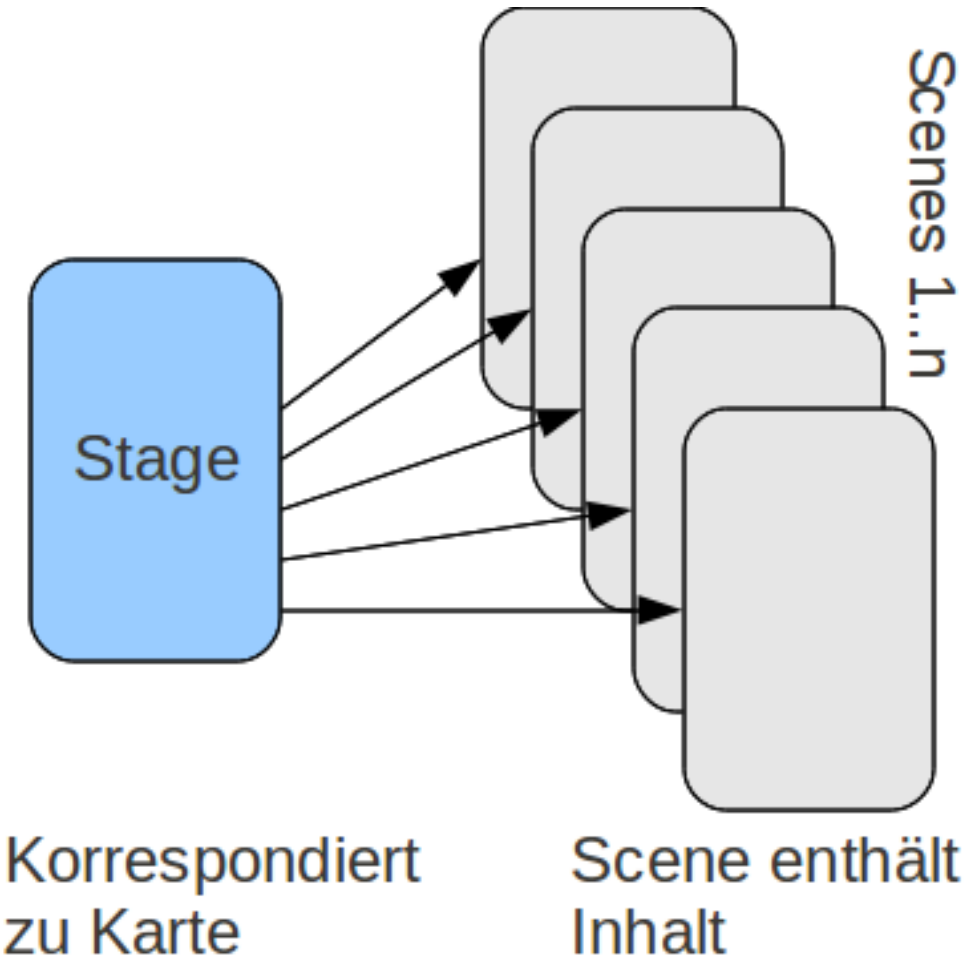
Inhaltsverzeichnis

- Übersicht
- Projekt Aufbau
- Hello World
- Arbeiten mit Mojo
  - Bauen, Installieren und Aufrufen
  - Testen
- Widgets
- Services
- Mensa App
- Fragen

WebOS Framework - Mojo

- stellt mehrere Scenes (Szenen) auf einer Stage dar
- eine Scene enthält verschiedene Widgets (Button, Liste, etc.)
- eine Scene und deren Elemente sind identisch zum W3C DOM
- Mojo enthält eine Services API

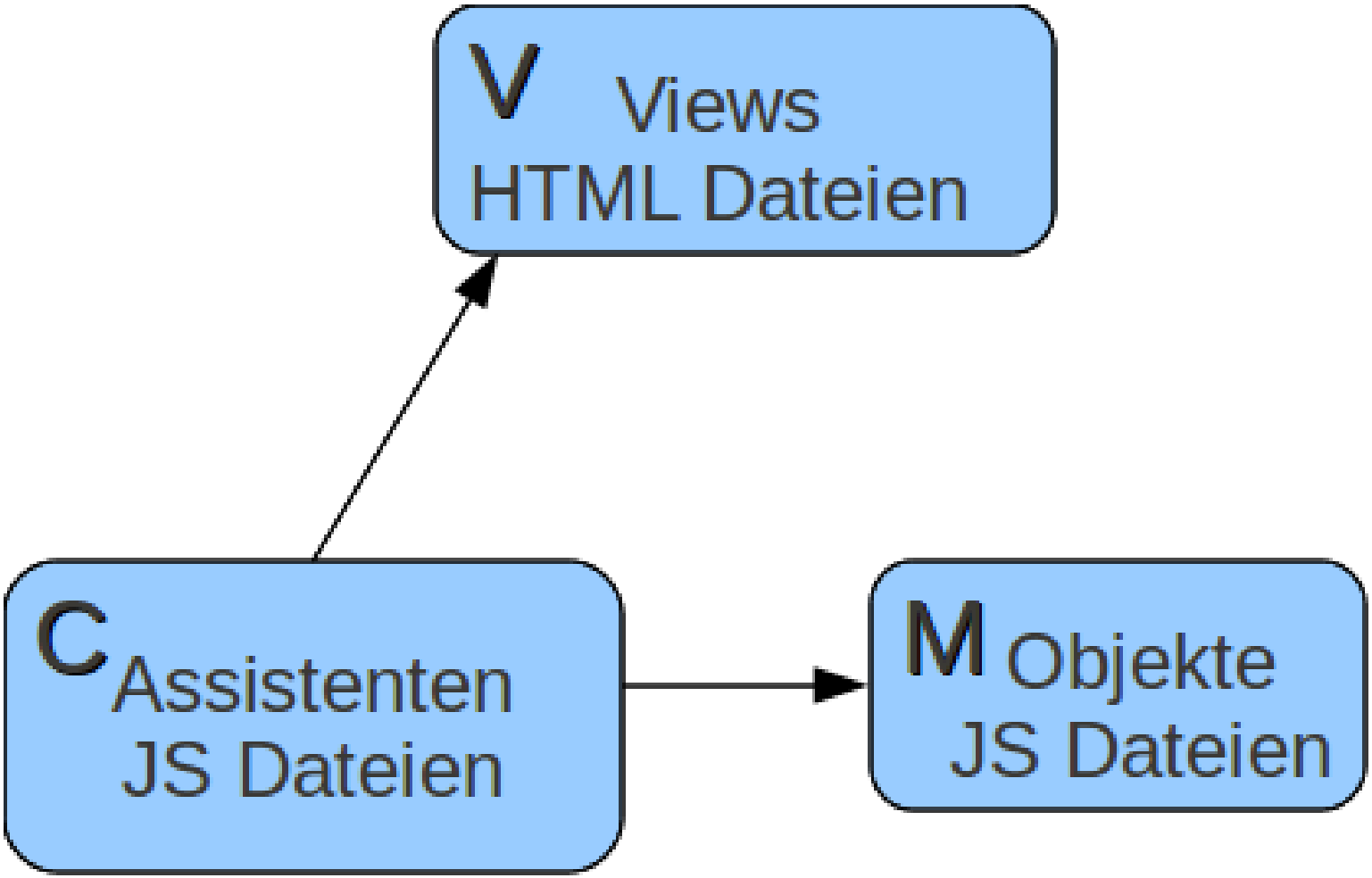
WebOS Framework - Mojo



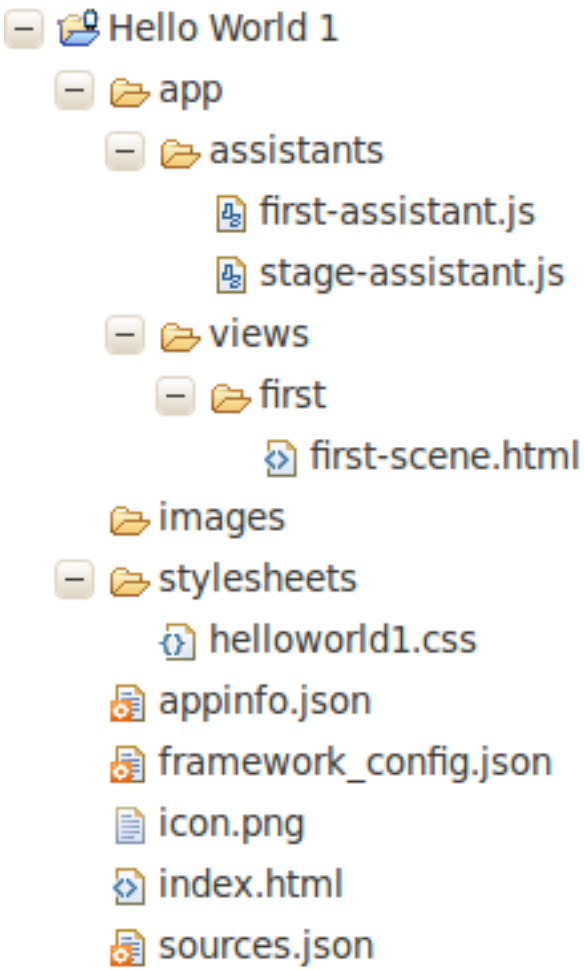
WebOS Framework - Mojo

- jede Scene, Stage und Anwendung besitzt einen Assistenten
- jeder Assistent besitzt eine Instanz auf seinen speziellen Type von Controller

MVC Muster


V
Views
HTML Dateien
C
Assistenten
JS Dateien
M
Objekte
JS Dateien

Mojo Projekt Struktur



sources.json

```
[
  {
    "source": "app/assistants/stage-assistant.js"
  },
  {
    "scenes": "first",
    "source": "app/assistants/first-assistant.js"
  }
]
```

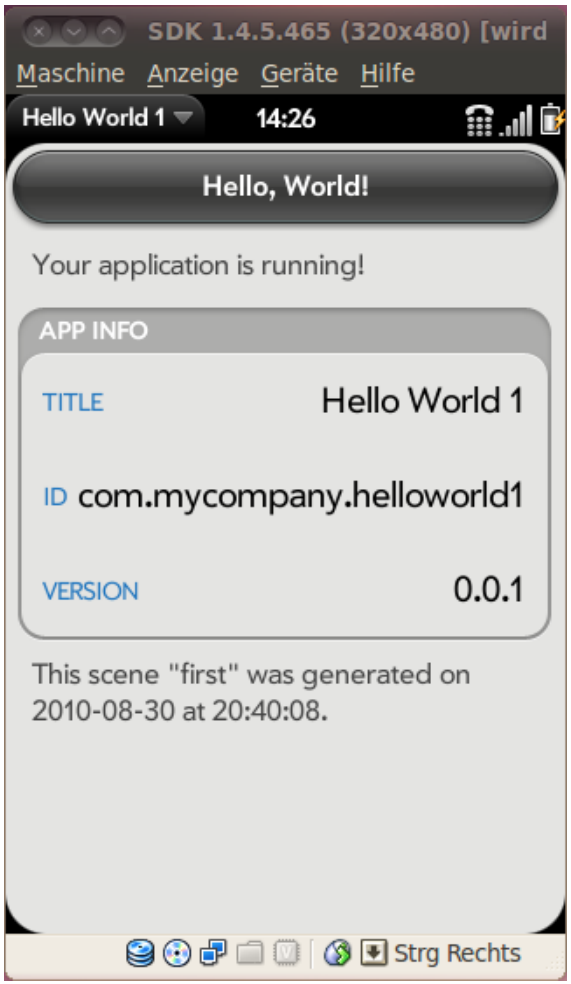
index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Hello World 1</title>
5  <script src="/usr/palm/frameworks/mojo/mojo.js"
6          type="text/javascript"
7          x-mojo-version="1"></script>
8
9  <!-- application stylesheet should come in
10         after the one loaded by the framework -->
11 <link href="stylesheets/helloworld1.css"
12         media="screen"
13         rel="stylesheet" type="text/css">
14 </head>
15 </html>
```

appinfo.json

```
1 {
2   "id": "com.mycompany.helloworld1",
3   "version": "0.0.1",
4   "vendor": "My Company",
5   "type": "web",
6   "main": "index.html",
7   "title": "Hello World 1",
8   "icon": "icon.png"
9 }
```

Hello World


SDK 1.4.5.465 (320x480) [wird
Maschine Anzeige Geräte Hilfe
Hello World 1
14:26
Hello, World!
Your application is running!
APP INFO
TITLE
Hello World 1
ID com.mycompany.helloworld1
VERSION
0.0.1
This scene "first" was generated on
2010-08-30 at 20:40:08.
Strg Rechts

first-scene.html - Teil 1

```
1 <div class="palm-hasheader">
2   <div class="palm-header">
3     Hello, World!
4   </div>
5 </div>
6 <div class="palm-header-spacer"></div>
7 <div class="palm-body-text">
8   Your application is running!
9 </div>
```

first-scene.html - Teil 2

```
1  <div class="palm-group">
2   <div class="palm-group-title">
3   <span x-mojo-loc="">APP INFO</span></div>
4   <div class="palm-list">
5    <div class='palm-row single'>
6      <div class="palm-row-wrapper">
7        <div class="label left">title</div>
8        <div id="app-title" class="title right">????</div>
9      <div class="palm-row-wrapper">
10       <div class="label left">id</div>
11       <div id="app-id" class="title right">????</div>
12     </div>
13     <div class="palm-row-wrapper">
14       <div class="label left">version</div>
15       <div id="app-version" class="title right">????</di
16     </div></div></div></div>
```

first-assistant.js

```
1  function FirstAssistant() {
2  }
3
4  FirstAssistant.prototype.setup = function() {
5  this.controller.get("app-title")
6     .update(Mojo.appInfo.title);
7  this.controller.get("app-id")
8     .update(Mojo.appInfo.id);
9  this.controller.get("app-version")
10    .update(Mojo.appInfo.version);
11 };
```

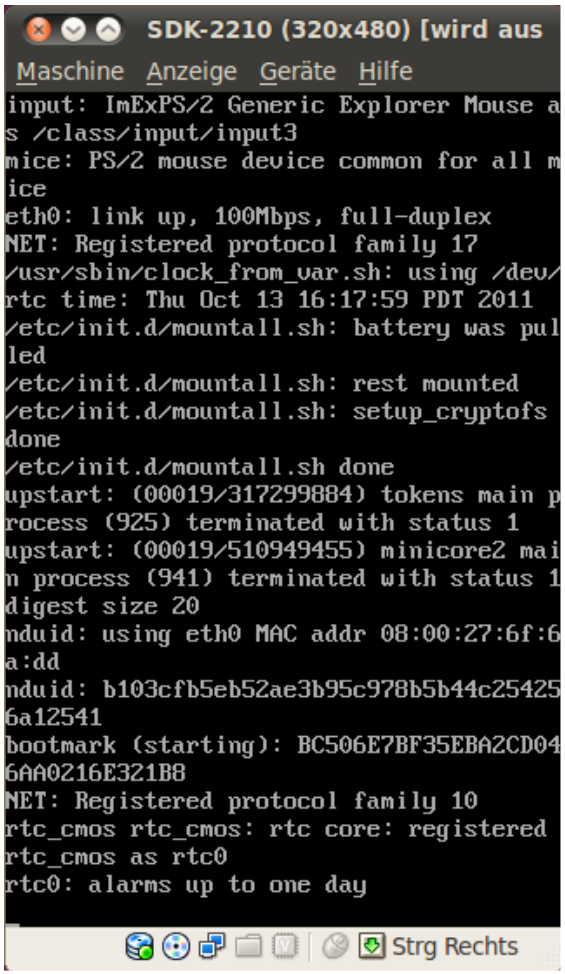
Bauen, Installieren und Aufrufen

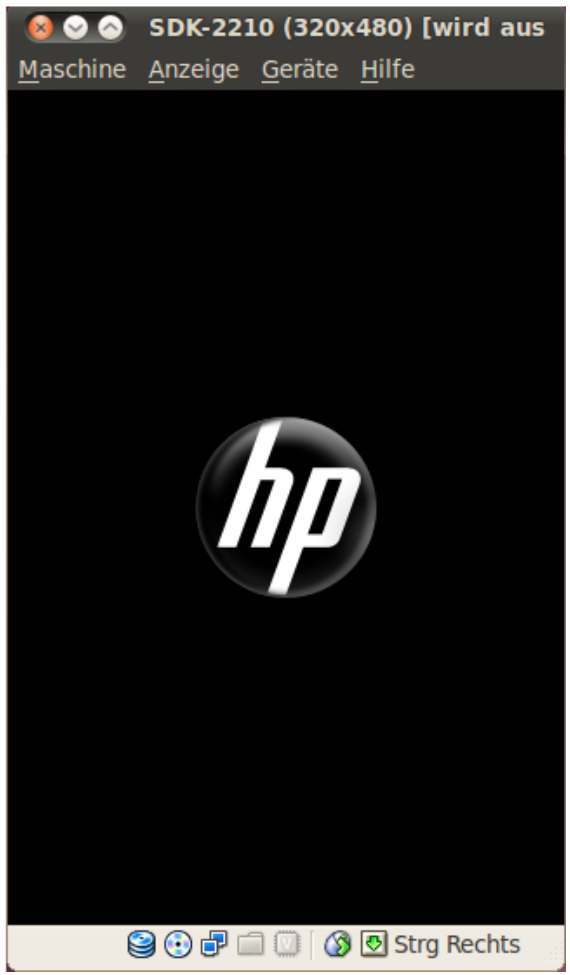
```
1 palm-package Hello\ World\ 1/
2 palm-install com.mycompany.helloworld1_0.0.1_all.ipk
3 palm-launch com.mycompany.helloworld1
```

Testen

- Anwendungen laufen in VirtualBox
- Debugging über Konsole (palm-log, novacom)

Testen


SDK-2210 (320x480) [wird aus
Maschine Anzeige Geräte Hilfe
input: ImExPS/2 Generic Explorer Mouse a
s /class/input/input3
mice: PS/2 mouse device common for all m
ice
eth0: link up, 100Mbps, full-duplex
NET: Registered protocol family 17
/usr/sbin/clock_from_var.sh: using /dev/
rtc time: Thu Oct 13 16:17:59 PDT 2011
/etc/init.d/mountall.sh: battery was pul
led
/etc/init.d/mountall.sh: rest mounted
/etc/init.d/mountall.sh: setup_cryptofs
done
/etc/init.d/mountall.sh done
upstart: (00019/317299884) tokens main p
rocess (925) terminated with status 1
upstart: (00019/510949455) minicore2 mai
n process (941) terminated with status 1
digest size 20
nduid: using eth0 MAC addr 08:00:27:6f:6
a:dd
nduid: b103cfb5eb52ae3b95c978b5b44c25425
6a12541
bootmark (starting): BC506E7BF35EBA2CD04
6AA0216E321B8
NET: Registered protocol family 10
rtc_cmos rtc_cmos: rtc core: registered
rtc_cmos as rtc0
rtc0: alarms up to one day
Strg Rechts


SDK-2210 (320x480) [wird aus
Maschine Anzeige Geräte Hilfe
Strg Rechts

Widgets

- Mojo.Widget.Button
- Mojo.Widget.CheckBox
- Mojo.Widget.DatePicker
- Mojo.Widget.Drawer
- Mojo.Widget.FilterField
- Mojo.Widget.FilterList
- Mojo.Widget.ImageView
- Mojo.Widget.IntegerPicker
- Mojo.Widget.List
- Mojo.Widget.ListSelector

Widget - View

```
1 <div id="myListSelector"
2       x-mojo-element="ListSelector"></div>
```

Widget - Assistant

```
1  var selectorChoices = [
2      {name:'Cosa', value:'spanish-thing'},
3      {name:'Chose', value:'french-thing'},
4      {name:'Ding', value:'german-thing'}
5  ];
6  var selectorAttributes = {
7      label: 'Pick a Thing',
8      choices: selectorChoices,
9      modelProperty:'value' };
10 this.selectorModel = {value:'spanish-thing'};
11 this.controller.setupWidget('myListSelector',
12       selectorAttributes, this.selectorModel);
```

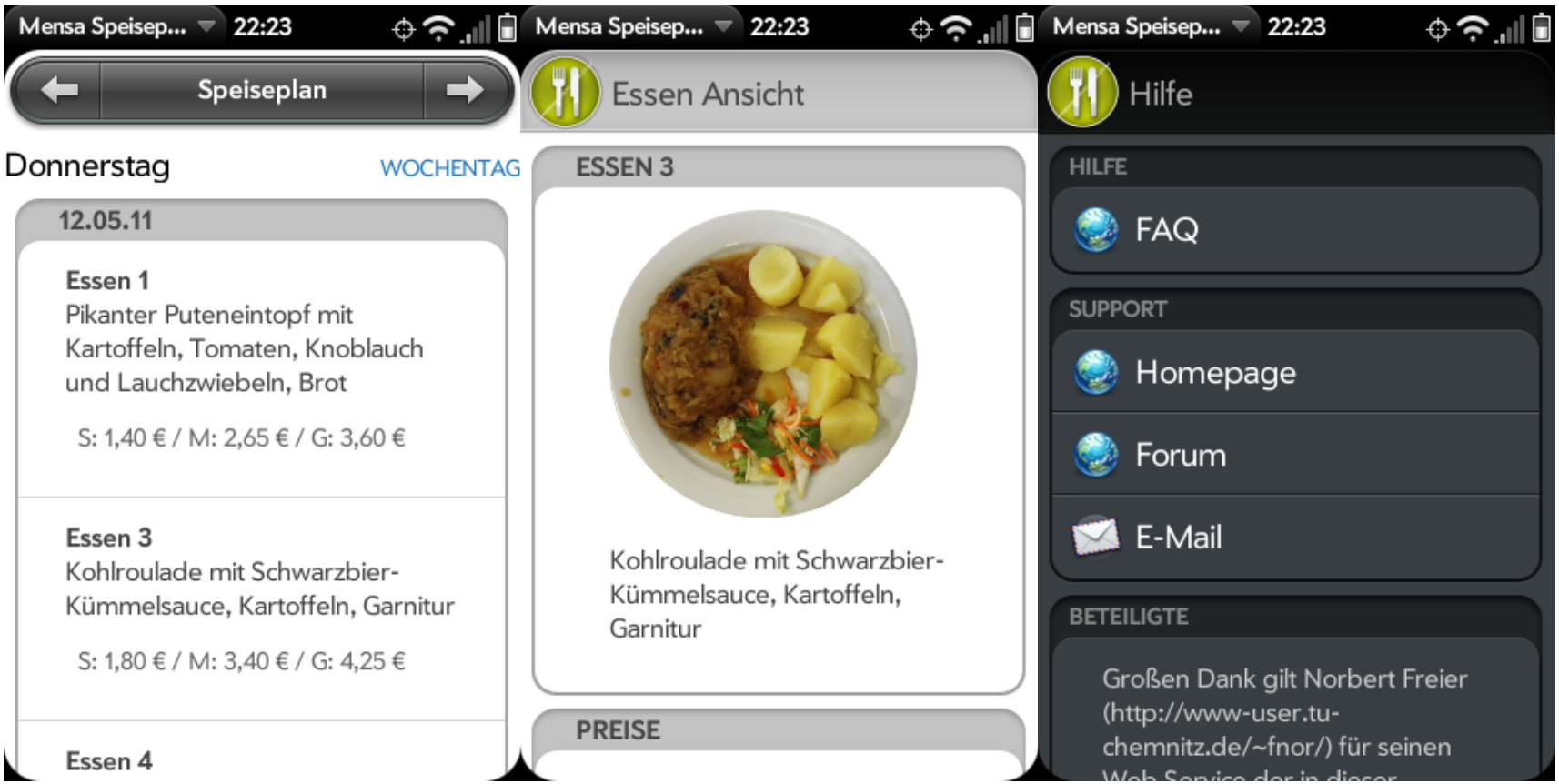
Services

- Bluetooth Serial Port Protocol (SPP)
- Connection Manager
- db8
- Display Manager
- Download Manager
- Firewall
- GPS
- In-App Payment
- Key Manager
- Key Service

Services - Beispiel

```
this.controller.serviceRequest(
    "palm://com.palm.applicationManager", {
        method: "open",
        parameters:  {
            id: 'com.palm.app.browser',
            params: {
                target: MensaSpeiseplan.supportForum
            }
        }
});
```

Mensa App


Mensa Speisep... 22:23
Speiseplan
Donnerstag
WOCHENTAG
12.05.11
Essen 1
Pikanter Puteneintopf mit Kartoffeln, Tomaten, Knoblauch und Lauchzwiebeln, Brot
S: 1,40 € / M: 2,65 € / G: 3,60 €
Essen 3
Kohlroulade mit Schwarzbier-Kümmelsauce, Kartoffeln, Garnitur
S: 1,80 € / M: 3,40 € / G: 4,25 €
Essen 4
Mensa Speisep... 22:23
Essen Ansicht
ESSEN 3
Kohlroulade mit Schwarzbier-Kümmelsauce, Kartoffeln, Garnitur
PREISE
Mensa Speisep... 22:23
Hilfe
HILFE
FAQ
SUPPORT
Homepage
Forum
E-Mail
BETEILIGTE
Großen Dank gilt Norbert Freier (http://www-user.tu-chemnitz.de/~fnor/) für seinen

Frage

Hast Du noch Fragen?

Teil 2

Entwickeln unter HP webOS

Das Framework Enyo

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

Inhaltsverzeichnis

Übersicht

Projekt Aufbau

Hello World

Arbeiten mit Enyo

- Bauen, Installieren und Aufrufen
- Testen

Mensa App

Fragen

WebOS Framework - Enyo

- basiert ebenfalls auf Javascript, HTML und CSS
- es existieren:
  - Controls
  - Components
  - Kinds

Enyo - Controls

- jedes Control wird in ein DOM Node transferiert

Aus

```
1 enyo.create({content: "Hello World"})
2       .renderInto(document.body);
```

wird

```
1 <div id="control">Hello World</div>
```

Enyo - Kinds

- entspricht dem Ansatz einer Klasse in Java
- unterstützt Vererbung

Enyo - Kinds

```
enyo.kind({
    name: "Point3D",
    kind: "Point",
    z: 0,
    constructor: function(x, y, z) {
        this.inherited(arguments);
        this.z = z;
    },
    translate: function(dx, dy, dz) {
        this.inherited(arguments);
        this.z += dz;
    },
    toString: function() {
        return this.inherited(arguments) + ", " + this.z;
    }
});
p = new Point3D(1, 1, 1);
```

Enyo - Components

- Components sind Grundbausteine von Enyo Apps
- eine Component enthält wiederum eine Liste von Components
- Event Handling

Enyo - Components Teil 1

```
1  enyo.kind({
2      name: "RandomizedTimer",
3      kind: enyo.Component,
4      baseInterval: 100,
5      percentTrigger: 50,
6      events: {
7          onTriggered: ""
8      },
9      create: function() {
10         this.inherited(arguments);
11         this.job = window.setInterval(
12             enyo.hitch(this, "timer"),
13             this.baseInterval);
14     },
```

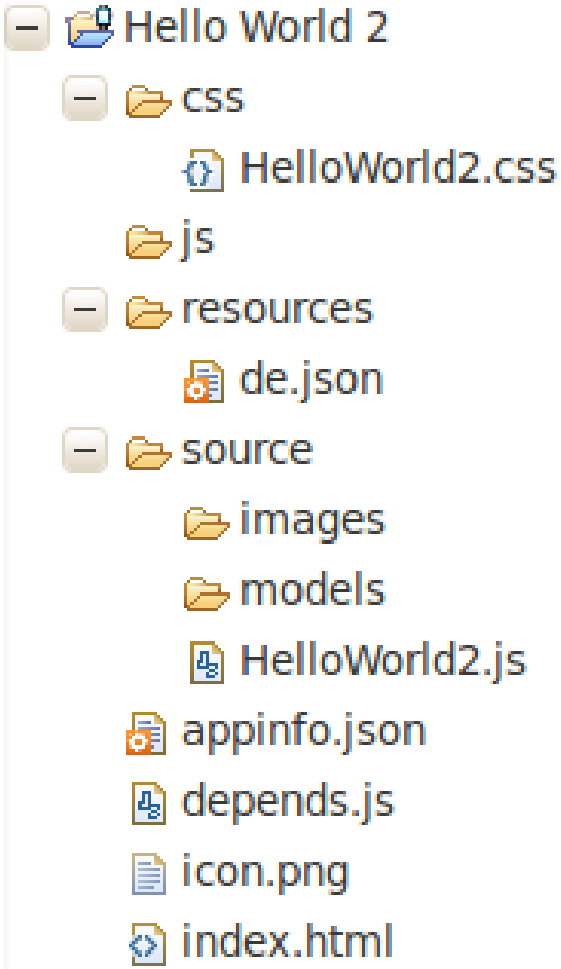
Enyo - Components Teil 2

```
1     destroy: function() {
2         window.clearInterval(this.job);
3     },
4     timer: function() {
5         if (Math.random() < this.percentTrigger * 0.01) {
6             this.doTriggered();
7         }
8     }
9 });
```

Enyo - Components Teil 3

```
1  enyo.kind({
2      name: "SimulatedMessage",
3      kind: enyo.Component,
4      components: [
5          {name: "timer", kind: RandomizedTimer,
6              percentTrigger: 10,
7              onTriggered: "timerTriggered"}
8      ],
9      timerTriggered: function() {
10         this.log("Simulated Service Message Occurred");
11     }
12 });
```

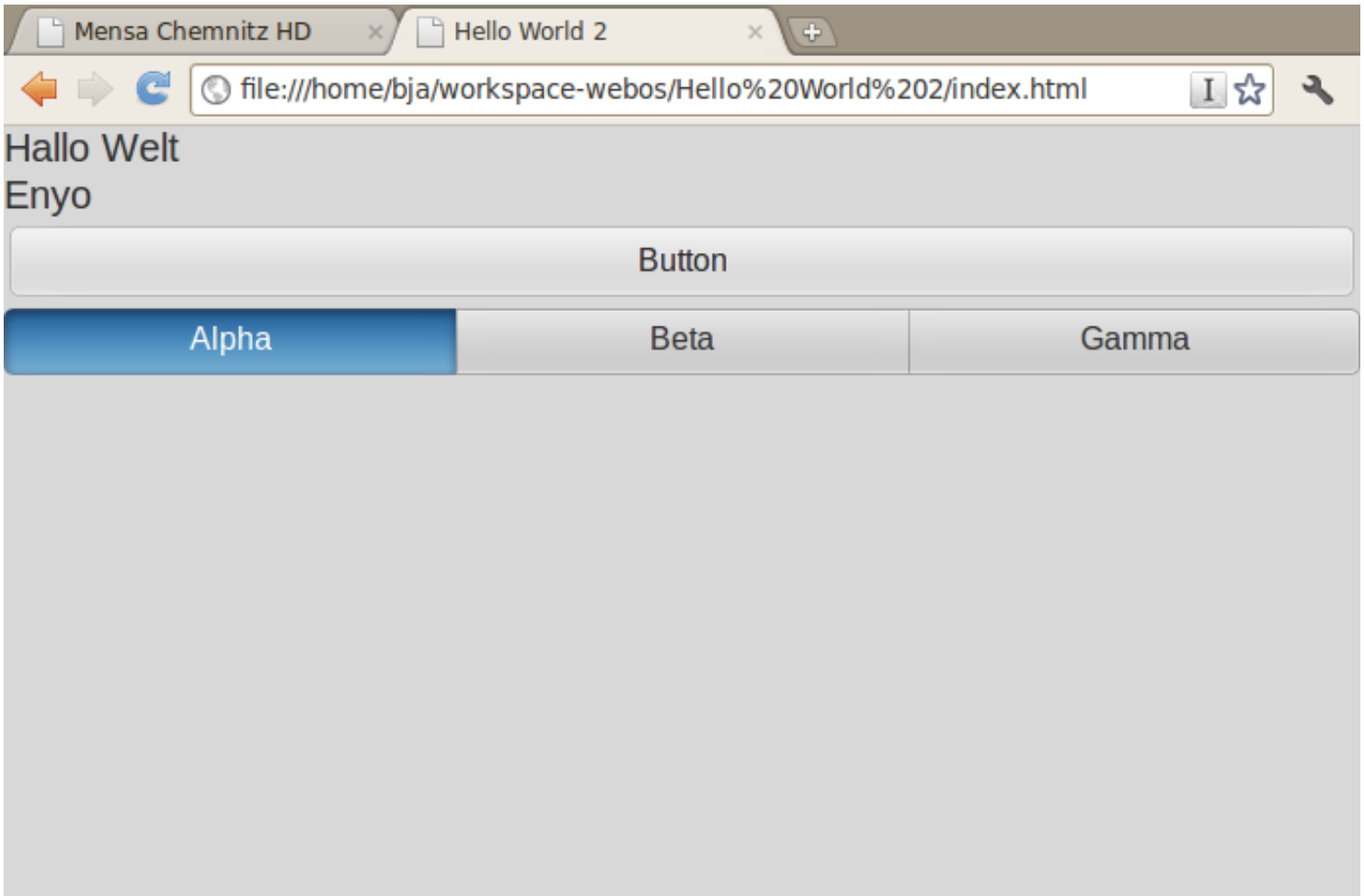
Enyo Projekt Struktur



HelloWorld2.js

```
1  enyo.kind({
2      name: "MyApps.HelloWorld2",
3      kind: enyo.VFlexBox,
4      components: [
5  {content: $L("Hello World")},
6  {content: "Enyo"},
7  {kind: "Button"},
8          // one-of-many selector with custom graphics
9          {kind: "RadioGroup", components: [
10             {label: "Alpha"},
11             {label: "Beta"},
12             {label: "Gamma"}
13         ]}
14     ]
15 });
```

Hello World


Mensa Chemnitz HD
Hello World 2
file:///home/bja/workspace-webos/Hello%20World%202/index.html
Hallo Welt
Enyo
Button
Alpha
Beta
Gamma

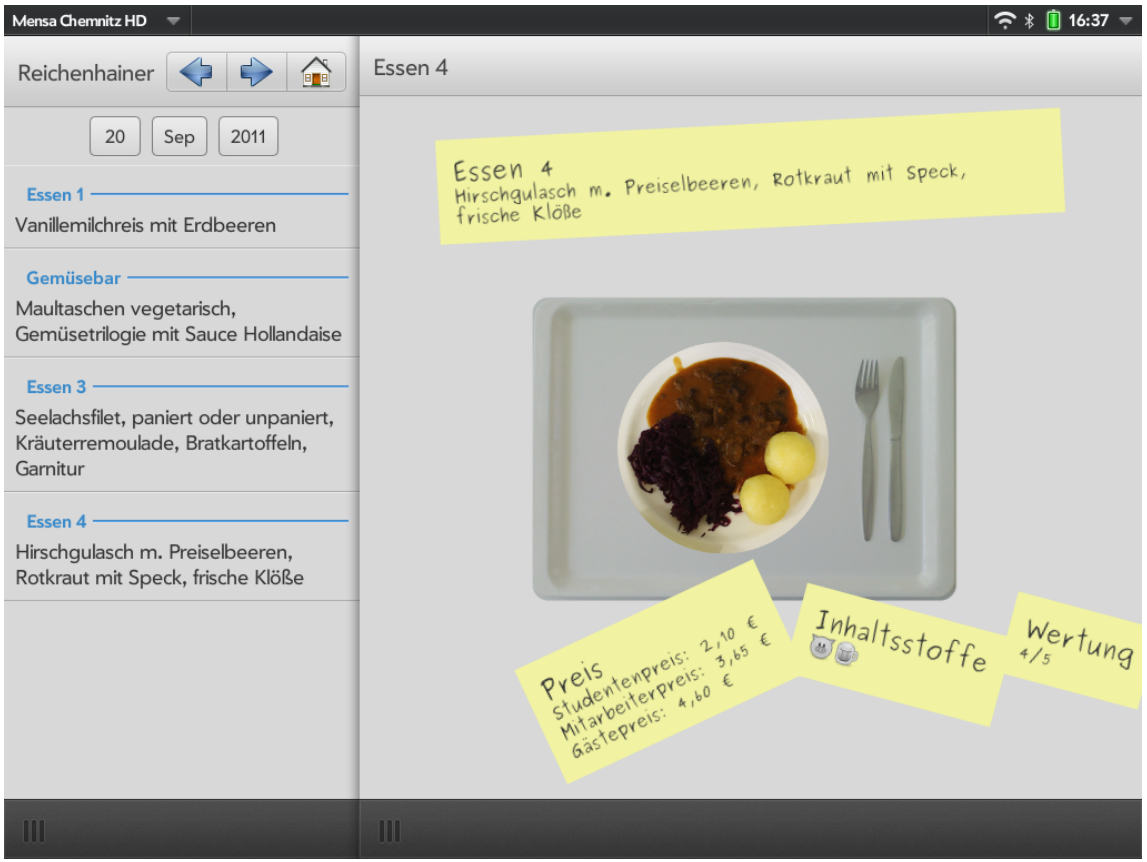
Bauen, Installieren und Aufrufen

```
palm-package Hello\ World\ 2/
palm-install com.mycompany.helloworld2_0.0.1_all.ipk
palm-launch com.mycompany.helloworld2
```

Testen

- Anwendungen laufen in VirtualBox
- Debugging über Konsole (palm-log, novacom)
- hauptsächliche Arbeit mit Chromium Browser

Mensa App


Mensa Chemnitz HD
16:37
Reichenhainer
20
Sep
2011
Essen 1
Vanillemilchreis mit Erdbeeren
Gemüsebar
Maultaschen vegetarisch, Gemüsetrilogie mit Sauce Hollandaise
Essen 3
Seelachsfilet, paniert oder unpaniert, Kräuterremoulade, Bratkartoffeln, Garnitur
Essen 4
Hirschgulasch m. Preiselbeeren, Rotkraut mit Speck, frische Klöße
Essen 4
Essen 4
Hirschgulasch m. Preiselbeeren, Rotkraut mit Speck, frische Klöße
Preis
Studentenpreis: 2,10 €
Mitarbeiterpreis: 3,65 €
Gästepreis: 4,60 €
Inhaltsstoffe
Wertung
4/5

Frage

Hast Du noch Fragen?

Literatur

- Mitch Allen: „Palm webOS“. O’Reilly, 2009.
- `http://www.precentral.net`
- `http://developer.palm.com`