Thursday, May 10, 2007,
Oracle, De Meern

Oracle Forms as Web Component

Date:
May 10, 2007

Location:
Oracle,
Rijnzathe 6,
De Meern,
The Netherlands

Time:
From 18:15 till 19:45

Wilfred van der Deijl

EUROTRANSPLANT

Introduction

- Wilfred van der Deijl
- Eurotransplant
  - Largest international Organ Procurement Organization
  - 7 European countries, 122 million inhabitants.
  - Based in Leiden, the Netherlands
  - 3 main tasks; Waiting List registration, Donor organ allocation, FollowUp

Background

- Want to migrate from Forms to ADF
  - 300+ Forms
  - 1280 remote users, 230 internal users
  - “Self service”/infrequent users
  - Intuitive user experience
- No big-bang scenario rewriting all Forms
- Use existing Forms in intuitive web application
  - Case-by-case
  - Smooth migration over time (ever 100% ?)

Key points

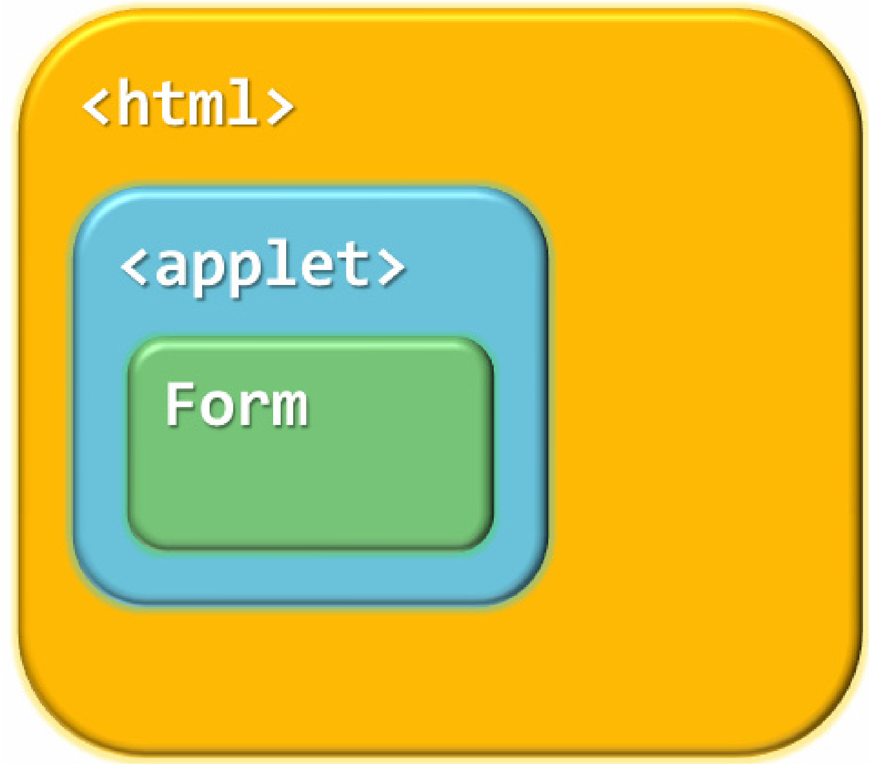
1. Pass context and events from JSF to Forms
2. Pass context and events from Forms to JSF
3. Application control from Forms to JSF
   - No inter-Form communication
   - Forms become individual components
   - Enable one-form-at-a-time migration
4. Eliminate Forms Applet startup time
5. Visual integration
   - Intuitive web user experience
   - Enables one-form-at-a-time migration

Demonstration

Including the Applet

Including the Applet

- Get HTML from basejpi.html or view Source
- Fill %-placeholders
- Absolute URLs
- Sun JVM, not Jinitiator
- Eolas patent dispute
- <af:regionDef> and <af:region>



EUROTRANSPLANT

oraFormsRegion.jspx

```
<?xml version='1.0' encoding='windows-1252'?>
<af:regionDef var="regionParams"
    xmlns:f="http://java.sun.com/jsf/
    xmlns:af="http://xmlns.oracle.com/adf/faces">
  <f:verbatim>
    . . . { actual HTML and JavaScript } . . .
  </f:verbatim>
</af:regionDef>
```

META-INF/region-metadata.xml

```
<component>
  <component-type>com.company...oraForm</component-type>
  <component-class>
    oracle.adf.view.faces.component.UIXRegion
  </component-class>
  <component-extension>
    <region-jsp-ui-def>
      /regions/oraFormsRegion.jspx
    </region-jsp-ui-def>
  </component-extension>
  <attribute>
    <attribute-name>formModuleName</attribute-name>
    <attribute-class>java.lang.String</attribute-class>
  </attribute>
</component>
```

Usage in ADF Faces page

```
<af:region id="oraFormRegion"
   regionType="com.company...oraForm">
  <f:attribute name="formModuleName“
               value="orders.fmx" />
</af:region>
```

Inbound JavaScript API

Two way communication

- Oracle OpenWorld September 2005
- Robin Zimmermann's concept
  - Pluggable Java Component (PJC) embedded in the Form
  - (invisible) custom applet on the same web page
  - Custom applet can expose public method
    - LiveConnect API
  - Custom applet and PJC set up socket communication on 127.0.0.1:port
- Simplified to our concept

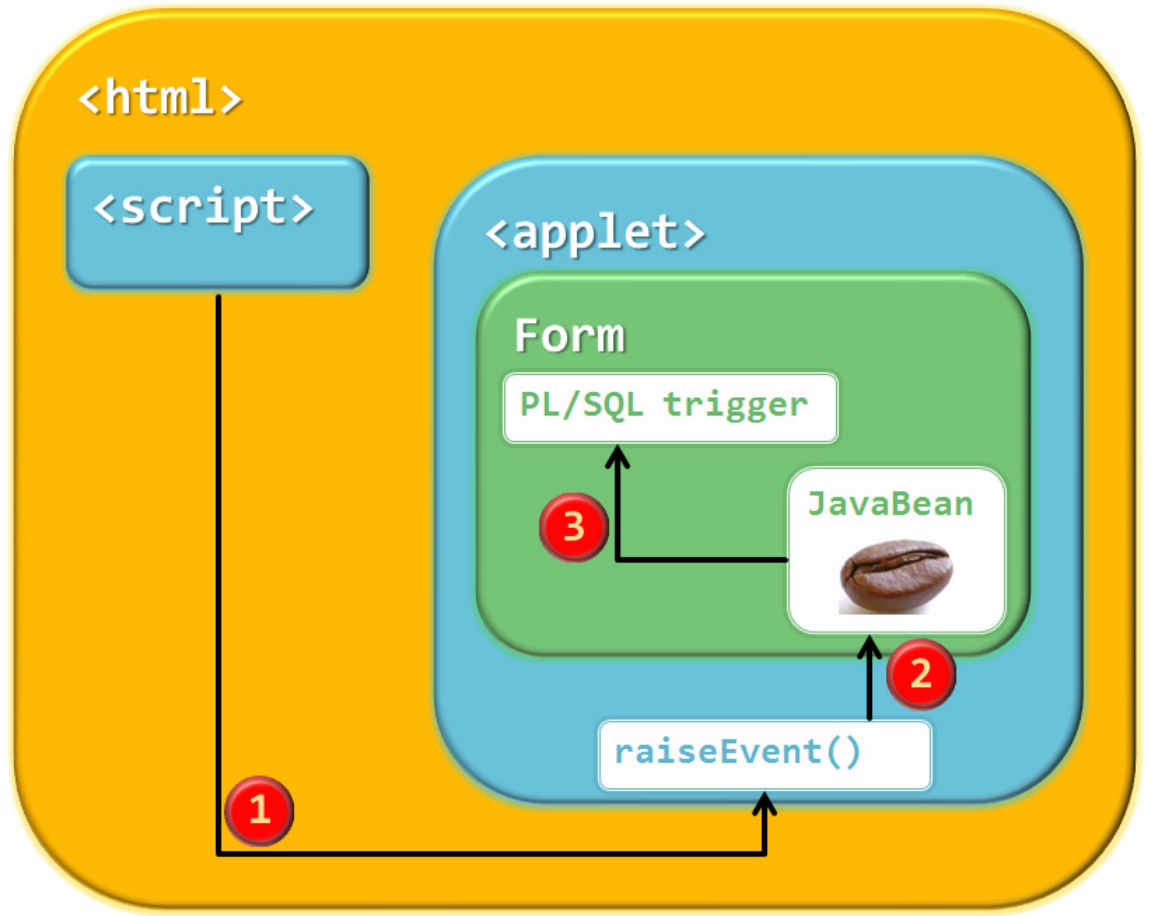
EUROTRANSPLANT

Inbound JavaScript API

- **Inbound** JavaScript API
- JSF button to raise Forms event (e.g. combined Save button)
- Foundation for other techniques
- Forms 11g will feature JavaScript API

Inbound JavaScript API

- Subclass and extend oracle.forms.engine.Main
- Package JAR
- Applet params
    - CODE
    - ARCHIVE



Sending message from JavaScript

```
document.formsapplet.raiseEvent('do_key',
                                'commit_form');
```

```
public void raiseEvent(String event, String payload) {
  CommunicatorBean communicator =
    findFirstCommunicator();
  if (null != communicator) {
    communicator.sendMessageToForms(event, payload);
  }
}
```

Message from PJC to Forms trigger

```
public void sendMessageToForms(String event,
                               String payload) {
  try {
    mHandler.setProperty(PROP_EVENT, event);
    mHandler.setProperty(PROP_PAYLOAD, payload);
    // trigger WHEN-CUSTOM-ITEM-EVENT trigger
    CustomEvent ce = new CustomEvent(mHandler,
                                 EVENT_MSG_TO_FORMS);
    dispatchCustomEvent(ce);
  } catch (FException e) {
    e.printStackTrace();
  }
}
```

Handle in Forms PL/SQL

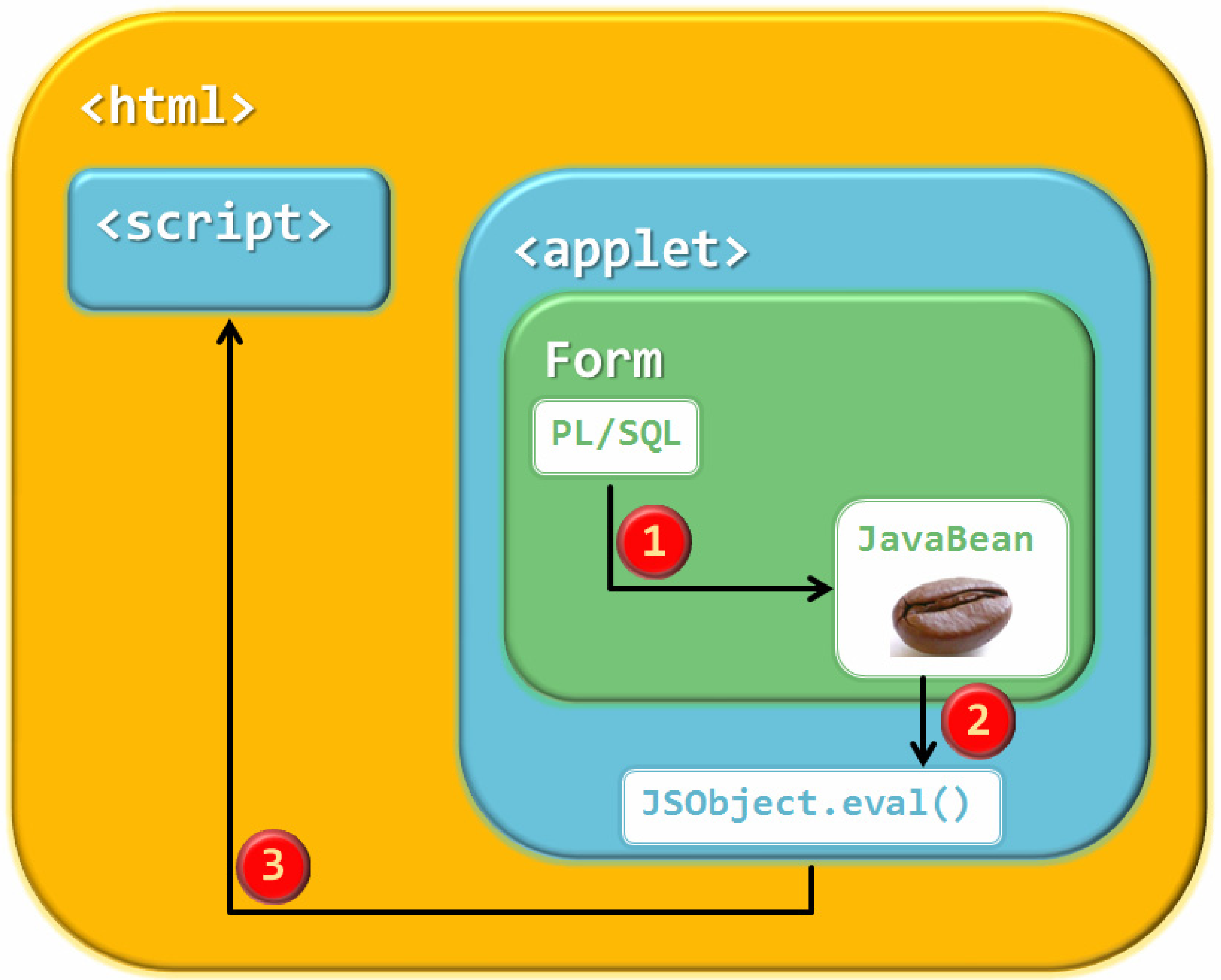
```
declare
  BeanEventDetails ParamList;
  ParamType        number := text_parameter;
  Event            varchar2(1000);
  Payload          varchar2(1000);
begin
  BeanEventDetails :=
    get_parameter_list(
      :system.custom_item_event_parameters);
  get_parameter_attr(BeanEventDetails, 'Event',
    ParamType, Event);
  get_parameter_attr(BeanEventDetails, 'Payload',
    ParamType, Payload);
  if event='do_key' then
    do_key(payload);
  end if;
end;
```

Outbound JavaScript API

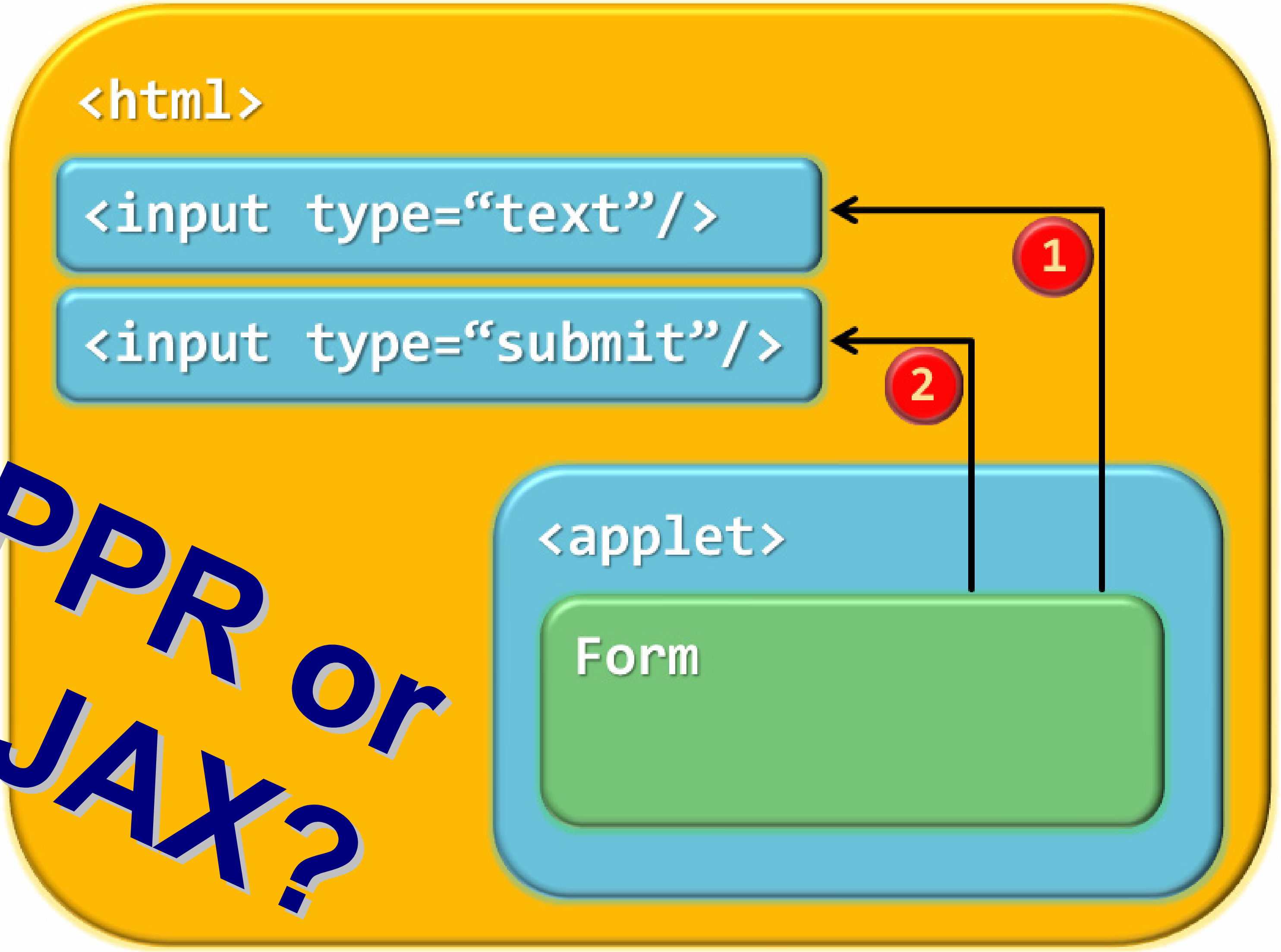
Outbound JavaScript API

- **Outbound** JavaScript API
- Forms to trigger events in web application
- Typical use: selected record in multi-record block
- Show details on same or other JSF page
- Use same PJC as Inbound JavaScript API
  - PJC can be “called” from PL/SQL
  - PJC can evaluate JavaScript API through applet

Outbound JavaScript API



Outbound JS API: set context



<html>

<input type=“text”/>

<input type=“submit”/>

<applet>

Form

PPR or AJAX?

Sending message from PL/SQL

```
begin
  -- set the Order ID
  set_custom_property('BLK_PJC.PJC', 1,
    'EvalExpression',
    'document.getElementById(''frm:ordid'')' ||
      '.value=' || :ord.ordid);
end;
```

Evaluating JavaScript

```
public boolean setProperty(ID property, Object
value) {
  if (PROP_EVAL_EXPR == property) {
    JSObject appletWindow =
      JSObject.getWindow(mHandler.getApplet());
    Object evalResult =
      appletWindow.eval(value.toString());
    // make result available as PJC property
    setProperty(PROP_EVAL_RESULT, evalResult);
    return true;
  }
}
```

Eliminate Applet startup time

Legacy Lifecycle Cache

- Applet startup resource and time consuming
- legacy_lifecycle = true
- Run in background
- Reactivate if 100% identical applet declaration
- Oracle Jinitiator 1.1.8 jinit_appletcache
- Sun JVM 1.4.2 or higher

EUROTRANSPLANT

Getting context into Forms

- Context: select ID (order, customer, …)
- Normally serverArgs → Forms parameter
- Legacy Lifecycle requires 100% identical applet
- Pull, not push parameters
- Use hidden input

```
<input type="hidden" id="ordernum"
  value="12345"/>
```

- Use Outbound JavaScript API to evaluate

```
document.getElementById('ordernum').value
```

EUROTRANSPLANT

When-New-Form-Instance

```
declare
  customerID  number;
begin
  -- get the customerID
  set_custom_property('BLK_PJC.PJC', 1,
    'EvalExpression',
    'document.getElementById(''frm:customerID'')' ||
    '.value');
  customerID := get_custom_property('BLK_PJC.PJC', 1,
                                    'EvalResult');
  :parameter.custid:=customerID;
  -- execute query (where uses :parameter.custid)
  do_key('execute_query');
end;
```

EUROTRANSPLANT

Need for a landing form

- Name of the form is a parameter
- Legacy Lifecycle requires 100% identical applet
- “Landing Form”
- When-New-Form-Instance gets name of form
    - Just as parameter: Outbound JavaScript API

When-New-Form-Instance

```
declare
  formName varchar2(1000);
begin
  -- get the form name
  set_custom_property('BLK_PJC.PJC', 1,
    'EvalExpression',
    'document.getElementById(' ||
    '''frm:oraFormRegion:formname'').value');
  formName := get_custom_property('BLK_PJC.PJC', 1,
                                  'EvalResult');
  -- start the form
  call_form(formName);
end;
```

Act on applet reactivation

- Applet resumes as it was when returning from Legacy Lifecycle Cache
- Add handling to re-activation
- Override deprecated show()
- Use Inbound JavaScript API route

```
    public void show() {
        super.show();
        raiseEvent("WHEN-APPLET-ACTIVATED", "");
      }
```

- Override stop() becomes unstable

When-New-Form-Instance

```
declare
  formName     varchar2(1000);
  AppletActive varchar2(10);
begin
  while true loop
    -- get the form name
    set_custom_property('BLK_PJC.PJC', 1, 'EvalExpression',
      'document.getElementById(''frm:oraFormRegion:formname''‘)' ||
      '.value');
    formName := get_custom_property('BLK_PJC.PJC', 1,
                                    'EvalResult');
    -- start the form
    call_form(formName);
    -- if we get here the calling form is closed
    AppletActive := get_custom_property('BLK_PJC.PJC', 1,
                                        'AppletActive');
    if (AppletActive='FALSE') then
    -- exit the loop if the user is closing the browser
      exit;
    end if;
  end loop;
end;
```

EUROTRANSPLANT

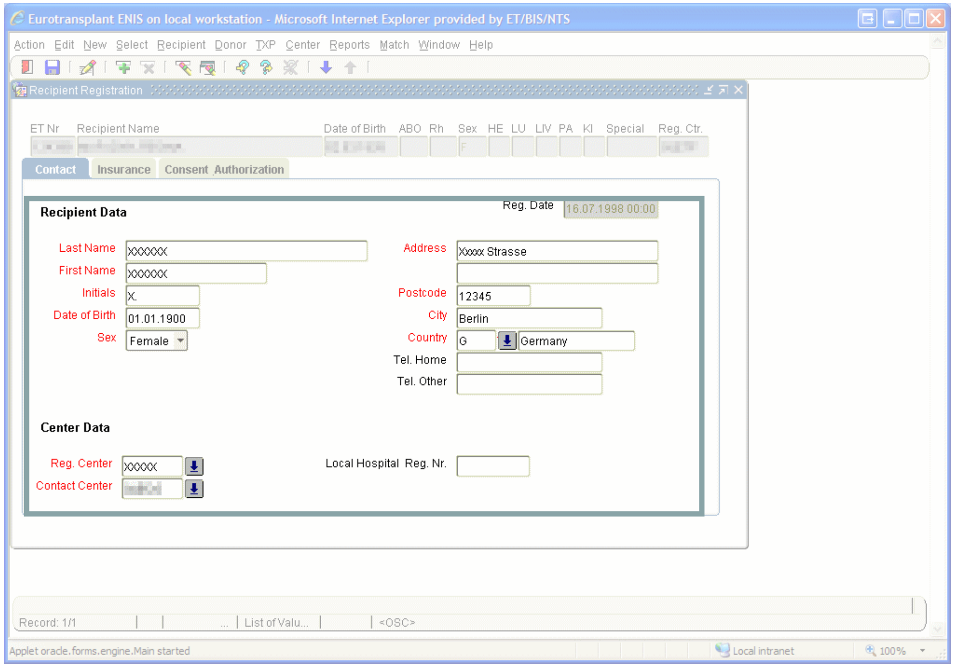
Visual Integration

Visual Integration

- Integrate seamlessly
  - User does not notice
  - Web-like intuitive user experience
  - One-form-at-a-time migration
- Remove chrome
  - Menu
  - Toolbar
  - Status bar
  - Forms (rounded) edges
  - Background color
  - Splash screen, logo, background image

Visual Integration

- Clip using surrounding <div>
  - <div> size smaller than applet
  - Applet negative relative X- and Y-pos
- No/minor changes to existing Forms
- Status bar invisible, but active



Eurotransplant ENIS on local workstation - Microsoft Internet Explorer provided by ET/BIS/NTS

Action Edit New Select Recipient Donor TXP Center Reports Match Window Help

Recipient Registration

ET Nr	Recipient Name	Date of Birth	ABO	Rh	Sex	HE	LU	LIV	PA	KI	Special	Reg. Ctr.
[redacted]	[redacted]	[redacted]			F							[redacted]

Contact
Insurance
Consent Authorization

Recipient Data

Reg. Date 16.07.1998 00:00

Last Name XXXXXX

First Name XXXXXX

Initials X.

Date of Birth 01.01.1900

Sex Female

Address Xxxxx Strasse

Postcode 12345

City Berlin

Country G Germany

Tel. Home

Tel. Other

Center Data

Reg. Center XXXXX

Contact Center [redacted]

Local Hospital Reg. Nr.

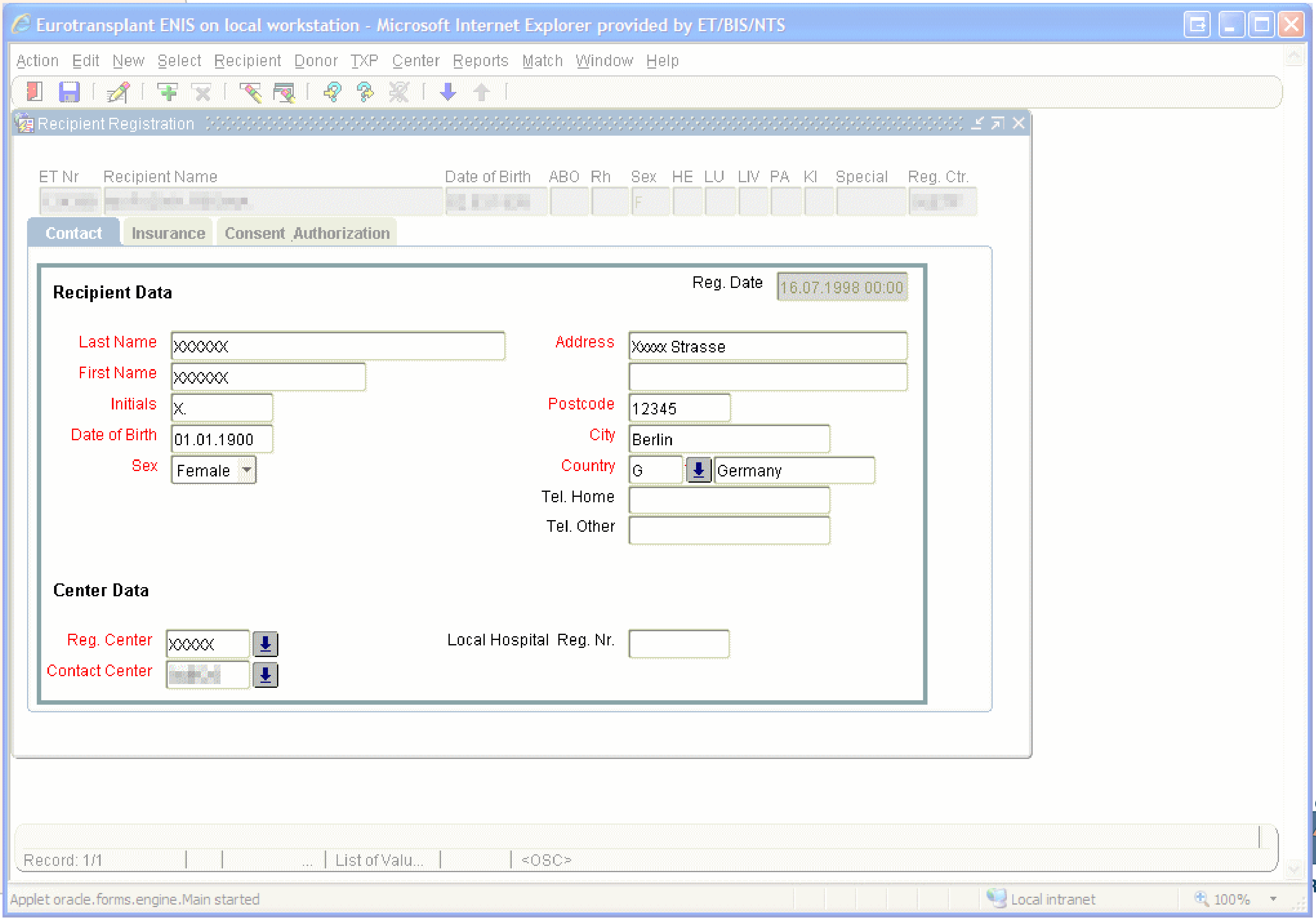
Record: 1/1 ... List of Valu... <OSC>

Applet oracle.forms.engine.Main started

Local intranet 100%

EUROTRANSPLANT

Clip Form with a <div> element



Eurotransplant ENIS on local workstation - Microsoft Internet Explorer provided by ET/BIS/NTS
Action Edit New Select Recipient Donor TXP Center Reports Match Window Help
Recipient Registration
Contact
Insurance
Consent Authorization
Record: 1/1
...
List of Valu...
<OSC>
Applet oracle.forms.engine.Main started
Local intranet
100%

ET Nr	Recipient Name	Date of Birth	ABO	Rh	Sex	HE	LU	LIV	PA	KI	Special	Reg. Ctr.
[redacted]	[redacted]	[redacted]			F							[redacted]

Recipient Data

Reg. Date 16.07.1998 00:00

Last Name	XXXXXX	Address	Xxxxx Strasse
First Name	XXXXXX		
Initials	X.	Postcode	12345
Date of Birth	01.01.1900	City	Berlin
Sex	Female	Country	G Germany
		Tel. Home	
		Tel. Other	

Center Data

Reg. Center	XXXXX	Local Hospital Reg. Nr.	
Contact Center	[redacted]		

RANSPLANT

Summary

- Forms can be integrated seamlessly
- Reuse of Forms as web application components
- JavaScript API similar to Forms 11g
- Integrate individual Forms or use it as a smooth transition from Forms to ADF stack
- Allows for Forms reuse in BPEL Human Task
- Eurotransplant will
  - Integrate all Forms initially
  - Immediate intuitive web user experience
  - Migrate to ADF over time (never 100%?)

License?

- Free to use
- Please do include:

```
public void show() {
  super.show();
  System.out.println("Concept of using Oracle
                      Forms as Web Component");
  System.out.println("courtesy of Wilfred van
                      der Deijl, OraTransplant");
  System.out.println("http://www.oratransplant.nl/
                 oracle-forms-as-web-component/");
}
```

EUROTRANSPLANT

Questions

- Please ask
- http://www.oratransplant.nl/

EUROTRANSPLANT