

DCGI

KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

WA 2

GWT

Martin Klíma

GWT – What is it?

Google Web Toolkig

- Compiler from Java to JavaScript + HTML
- Set of JavaScript and Java scripts / classes
- Development environment – SDK
- Integration with IDE – Eclipse, Netbeans, ...

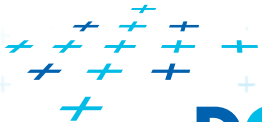
Driving idea:

Let's code everything in just one language– Java

Client side will be translated as needed

Write the GUI in a Swing style

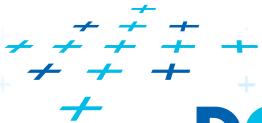
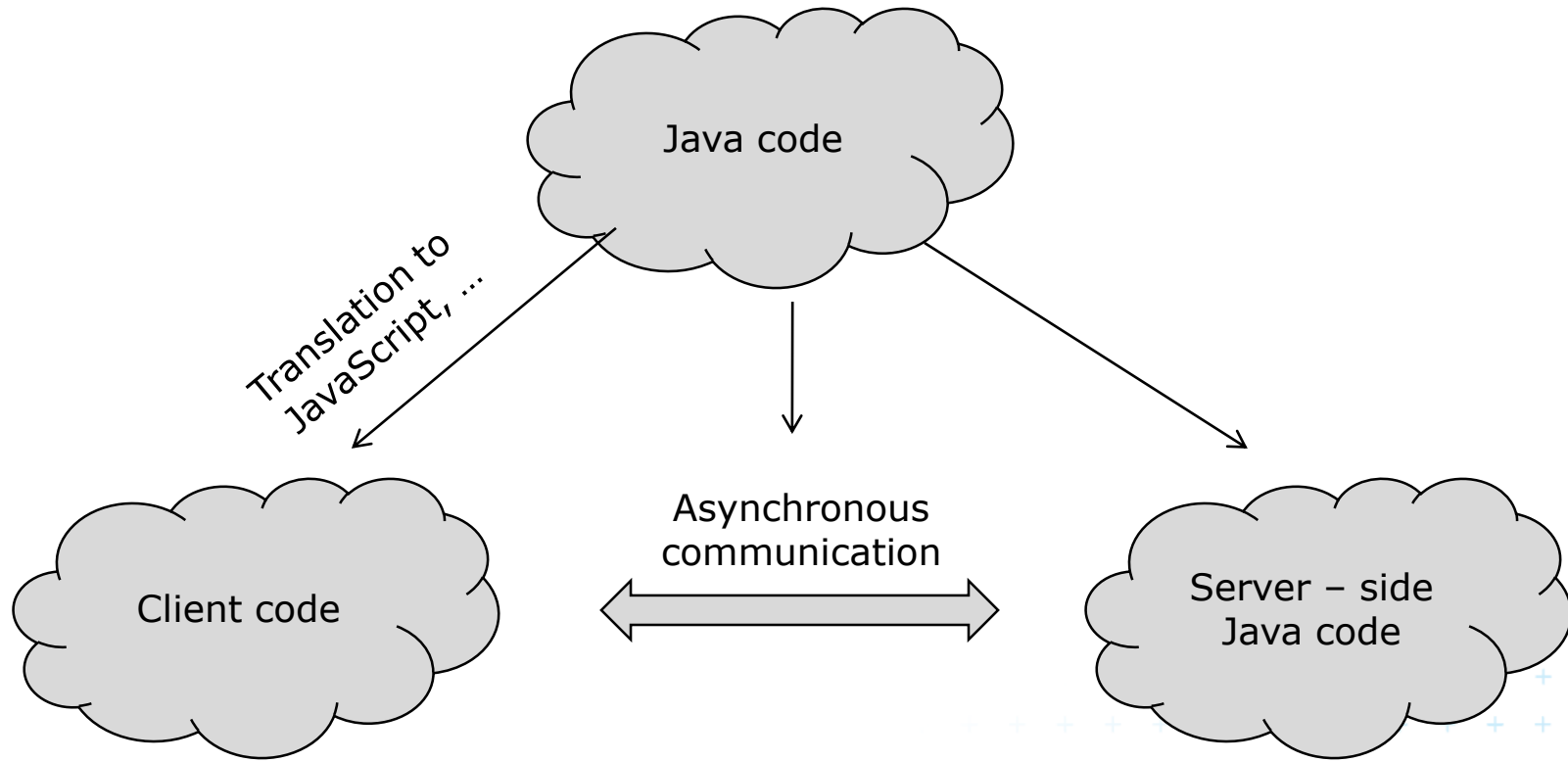
Use the classical event handlers in GUI



DCGI

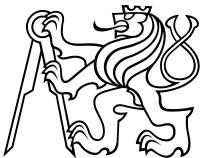


Architecture – AJAX in the main role



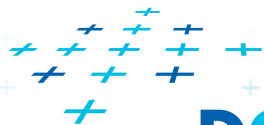
GWT

- GWT is pronounced “gwit”
- Open source project with Apache Licence 2.0
- Created by Google
- Many inspired and derived projects
 - Vaadin
 - Smart GWT
 - Sencha GXT
- IDE integration GWT4NB – for NetBeans
- Links:
 - www.gwtproject.org

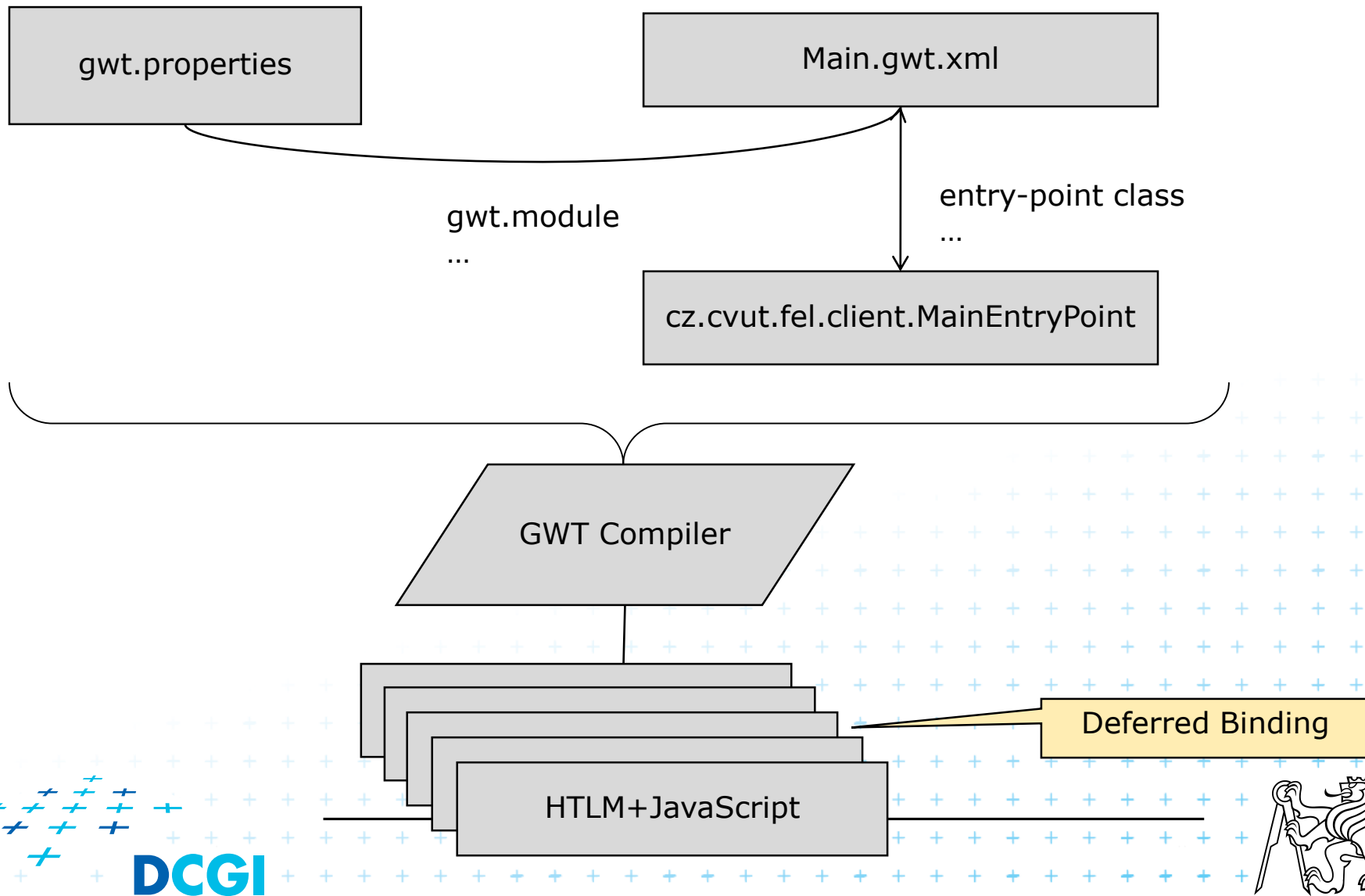


GWT Modules

- Project divided to models
- A module is a coherent functionality
- Java like naming convention
- Each module has a descriptor in XML `.gwt.xml`
 - Inheritance
 - Entry-point class
 - Source paths
 - classes will be translated to JavaScript
 - Class paths
 - External JavaScript
 - CSS
 - Servlets, other „properties“



How does it work?



GWT Properties

```
# The name of the module to compile
gwt.module=cz.cvut.fel.dama

# Folder within the web app context path where the output
# of the GWT module compilation will be stored.
# This setting is only used for GWT 1.5. For newer versions please
# use
# the rename-to attribute in the GWT module file (.gwt.xml).
gwt.output.dir=/cz.cvut.fel.dama

# Script output style: OBF[USCATED], PRETTY, or DETAILED
gwt.compiler.output.style=OBF

# Additional JVM arguments for the GWT compiler
gwt.compiler.jvmargs=-Xmx256M

# Specifies the number of local workers to use when compiling
# permutations and module(s)
gwt.compiler.local.workers=1

# The level of logging detail: ERROR, WARN, INFO, TRACE,
# DEBUG,
gwt.compiler.logLevel=WARN
```

```
# Script output style: OBF[USCATED], PRETTY, or DETAILED
gwt.shell.output.style=OBF

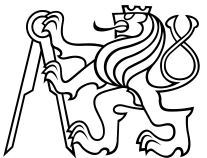
# The level of logging detail: ERROR, WARN, INFO, TRACE,
# DEBUG,
gwt.shell.logLevel=WARN

# Additional JVM arguments for the GWT shell/GWT hosted mode
# (GWT 1.6)
# Add -d32 here and use at least GWT 1.7.1 to debug on a Mac
# (32-bit JRE is required by GWT for debugging)
gwt.shell.jvmargs=-Xmx256M

# GWT version: 1.5,1.6,1.7 or 2.0
gwt.version=2.0

# GWT 2.0 only
# Specifies the TCP port for the code server
gwt.shell.code.server.port=9997

# GWT 2.0 only
# Specifies the TCP port for the embedded web server
gwt.shell.port=8888
```



GWT - module entry point

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE module PUBLIC "-//Google Inc.//DTD Google Web Toolkit 1.7.0//EN"
"http://google-web-toolkit.googlecode.com/svn/tags/1.7.0/distro-source/core/src/gwt-module.dtd">
```

```
<module>
```

```
<inherits name="com.google.gwt.user.User"/>
```

```
<inherits name='com.google.gwt.user.theme.standard.Standard'/>
```

```
<!-- Inherit the default GWT style sheet. You can change -->
```

```
<!-- the theme of your GWT application by uncommenting -->
```

```
<!-- any one of the following lines. -->
```

```
<!-- <inherits name='com.google.gwt.user.theme.standard.Standard'/> -->
```

```
<!-- <inherits name="com.google.gwt.user.theme.chrome.Chrome"/> -->
```

```
<!-- <inherits name="com.google.gwt.user.theme.dark.Dark"/> -->
```

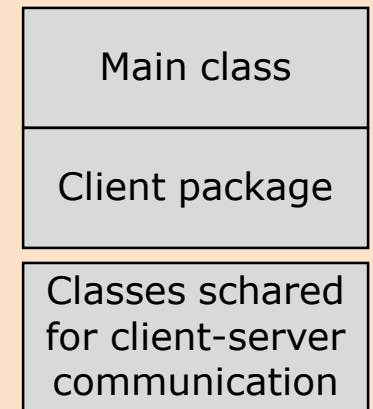
```
<entry-point class="cz.cvut.fel.client.damaEntryPoint"/>
```

```
<source path="client"/>
```

```
<source path="shared"/>
```

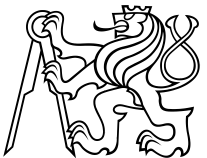
```
<!-- Do not define servlets here, use web.xml -->
```

```
</module>
```



Class loading

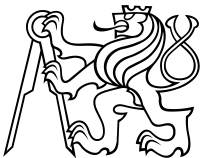
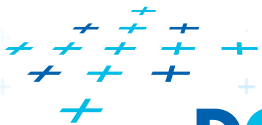
1. Browser loads the starting HTML page
2. Will interpret the script tag
`<script src="<Module Name>.nocache.js">` , and load the JavaScript
3. In JavaScript there is the deferred binding condition leading to loading particular .cache.html
4. .cache.html creates a hidden `<iframe>` and loads into it
5. After loading .cache.html the main application code is started



Loading HTML page example

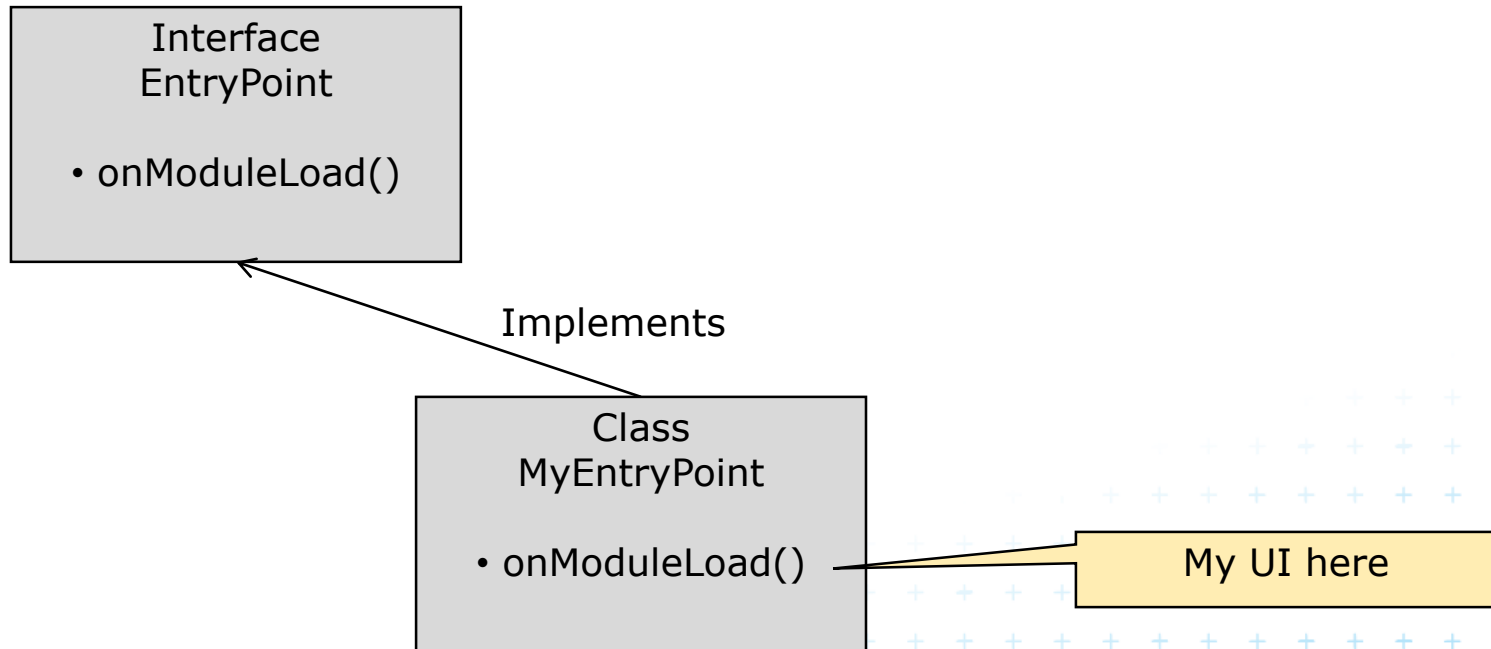
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <link type="text/css" rel="stylesheet" href="dama.css">
    <title>Dáma na webu</title>
    <!-- Nahrání klientské logiky v javascriptu -->
    <script type="text/javascript" language="javascript" src="cz.cvut.fel.dama/cz.cvut.fel.dama.nocache.js"></script>
  </head>
  <body>
    <!-- iframe pro podporu historie -->
    <iframe src="javascript:\"" id="__gwt_historyFrame" tabIndex='-1'
style="position:absolute;width:0;height:0;border:0"></iframe>

    <h1>Webová dáma</h1>
    <div id="ovladani"></div>
    <div id="sachovnice"></div>
  </body>
</html>
```



Example – client code

- The entry point is a java class implementing EntryPoint interface



Implementation

```
package cz.cvut.fel.client;  
  
import com.google.gwt.core.client.EntryPoint;  
import com.google.gwt.user.client.ui.Button;  
import com.google.gwt.user.client.ui.Label;  
import com.google.gwt.user.client.ui.RootPanel;  
import com.google.gwt.event.dom.client.ClickEvent;  
import com.google.gwt.event.dom.client.ClickHandler;
```

```
public class MainEntryPoint implements EntryPoint {
```

```
    public MainEntryPoint() { }
```

```
    public void onModuleLoad() {  
        final Label label = new Label("Hello, GWT!!!");  
        final Button button = new Button("Click me!");
```

```
        button.addClickHandler(new ClickHandler() {  
            public void onClick(ClickEvent event) {  
                label.setVisible(!label.isVisible());  
            }  
        });
```

```
        RootPanel.get().add(button);  
        RootPanel.get().add(label);
```

```
    }
```

```
}
```

Code is similar to Swing

UI on the client side driven by events

USER INTERFACE



UI similar to Swing

RootPanel – container of other components

FlowPanel – simple <div> on the page

HTMLPanel – builds HTML structure

FormPanel – a form

ScrollPanel – panel with scroll

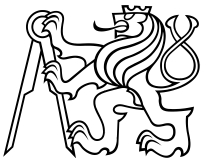
PopupPanel, DialogBox - dialogs

Grid, FlexTable – tables

LayoutPanel, DockLayoutPanel, SplitLayoutPanel,
StackLayoutPanel, TabLayoutPanel,

More info here:

<http://www.gwtproject.org/javadoc/latest/>



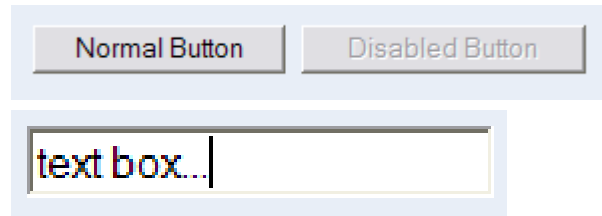
Attention

GWT layout works well only in standard browser mode, avoid quirk mode.



GUI components

- Button



- TextBox

☐ 🏠 foo@example.com

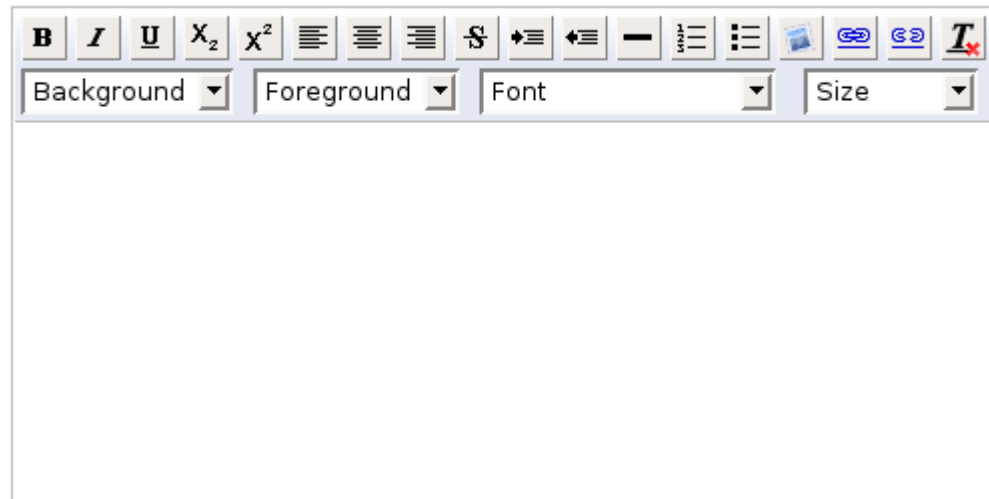
✉ Inbox

📄 Drafts

📁 Templates

- Tree

- RichTextArea



Events

```
public void anonClickHandlerExample() {  
    Button b = new Button("Click Me");  
    b.addClickHandler(new ClickHandler() {  
  
        public void onClick(ClickEvent event) {  
            // handle the click event  
        }  
    });  
}
```

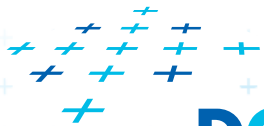
Anonymous
internal class

attention – use the *final*
modifier

```
class MojeTrida implements ClickHandler {  
    public void onClick(ClickEvent event) {  
        // tady si odchytnu udalost a udelam s ni co potrebuji  
    }  
}
```



SERVER COMMUNICATION



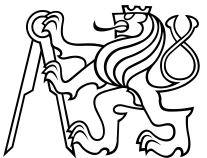
DCGI



GWT - RPC

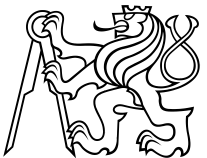
■ RPC = Remote Procedure Call

- calling server code from client code
- Client – JavaScript
- Server – Servlet
- Asynchronous communication using AJAX
- Serialized objects mediate the communication
 - *char, byte, short, int, long, boolean, float, double*
 - *String, Date, Character, Byte, Short, Integer, Long, Boolean, Float, Double*
 - array of serialized values
 - serialized class (*IsSerializable* or *java.io.Serializable*)
 - a class having at least one serializable subclass



GWT – RPC cont.

- Synchronous interface
- Asynchronous interface
- Implementation of asynchronous interface on server = **contract**
- Calling the asynchronous interface by java code from the client
- Implementation of synchronous interface by the server



GWT - RPC

Client

Server

GWT

- PozpatkuServiceAsync create (Class trida)

Synchronní rozhraní

- String pozpatku(string s)

RemoteServiceServlet

- processCall(String s)
- ...

Client code

```
final PozpatkuServiceAsync pozpatkuServiceAsync =
    GWT.create(PozpatkuService.class);

pozpatkuServiceAsync.pozpatku(textbox.getText(), new
AsyncCallback<String>() {
    public void onFailure(Throwable caught) {
        // udelej neco pri chybe
    }
    public void onSuccess(String result) {
        // udelame neco s vracenym retezcem
    }
});
```

Server code

```
public class PozpatkuServiceImpl extends
RemoteServiceServlet
implements PozpatkuService {

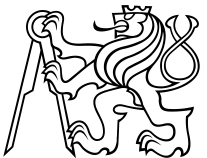
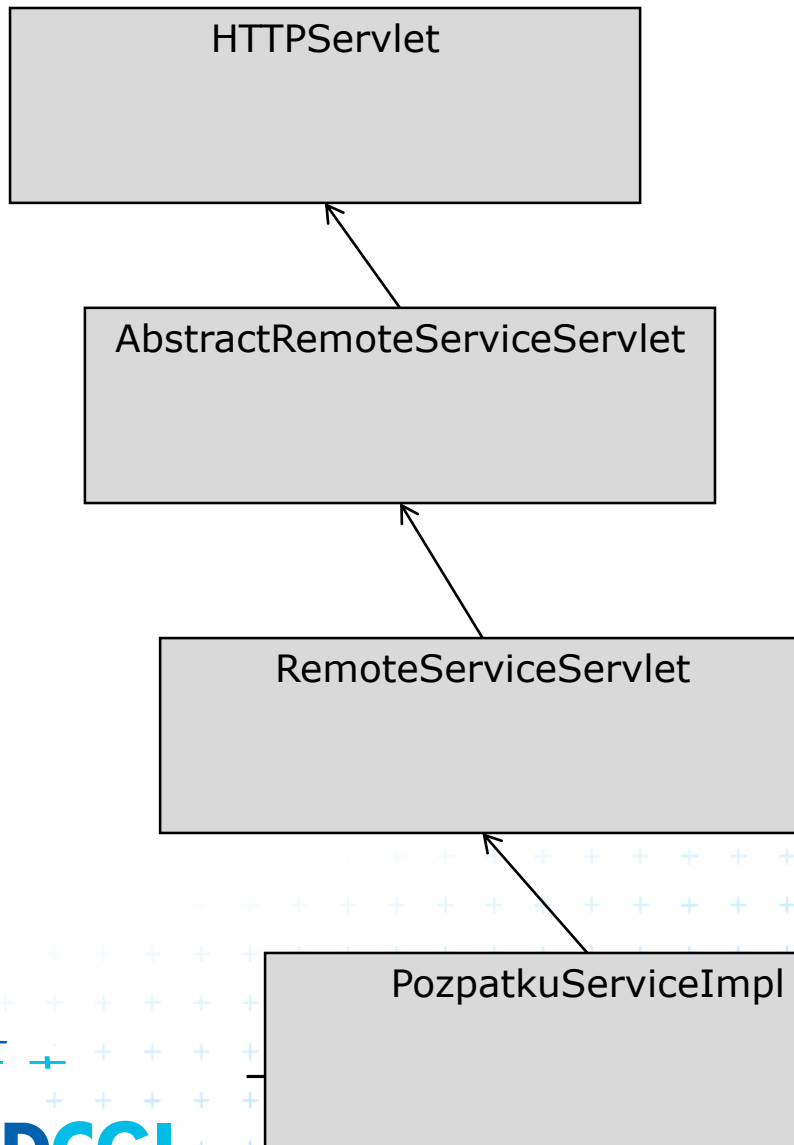
    public String pozpatku(String s) {
        StringBuffer reverse =
            new StringBuffer(s.length());
        for (int i = (s.length() - 1); i >= 0; i--) {
            reverse.append(s.charAt(i));
        }
        return reverse.toString();
    }
}
```

Application state

- By Ajax the application state is stored by the client
- Refresh page problem
- Must backup the client state at the server side



Sessions



Working with the session on the server

```
public class PozpatkuServiceImpl extends RemoteServiceServlet implements PozpatkuService {  
  
    public String pozpatku(String s) {  
        StringBuffer reverse = new StringBuffer(s.length());  
        for (int i = (s.length() - 1); i >= 0; i--) {  
            reverse.append(s.charAt(i));  
        }  
        String toReturn = reverse.toString();  
        // ulozim si posledni string do session  
        getThreadLocalRequest().getSession().setAttribute("lastString", toReturn);  
        return toReturn;  
    }  
  
    public String getLast() {  
        return (String) getThreadLocalRequest().getSession().getAttribute("lastString");  
    }  
}
```

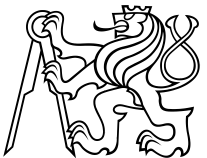
Set the session

Reading the



GWT and history in browser

- Problem: Ajax based application do not generate standard history in the browser
- Solution in GWT: iframe will store the history on command
- How it works:
 1. We insert an iFrame into HTML
 2. Based on event X we actively record the history as a String
 3. We react on event of transition in history by reading the strings, decoding the history and setting a new state.

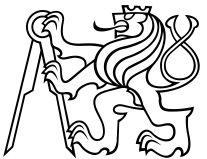


Insert a frame to HTML iframe

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta name='gwt:module' content='cz.cvut.fel.Main=cz.cvut.fel.Main'>
    <title>Main</title>
  </head>
  <body>
    <script type="text/javascript" src="cz.cvut.fel.Main/cz.cvut.fel.Main.nocache.js"></script>
    <!-- include this if you want history support -->
    <iframe id="__gwt_historyFrame" style="width:0;height:0;border:0"></iframe>
  </body>
</html>
```

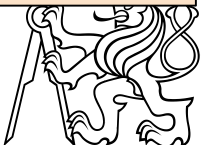
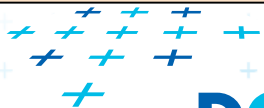
Write the history description to the record

```
textbox.addValueChangeListener(new ValueChangeListener<String>() {  
  
    public void onValueChange(ValueChangeEvent<String> event) {  
        History.newItem("text_" + event.getValue());  
    }  
});
```

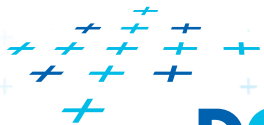


Use the history description, set application state

```
History.addValueChangeListener(new ValueChangeListener<String>() {  
  
    public void onValueChange(ValueChangeEvent<String> event) {  
        String historyToken = event.getValue();  
  
        // Parse the history token  
  
        if (historyToken.substring(0, 5).equals("text_")) {  
            String historyString =  
                historyToken.substring(5, historyToken.length());  
            // Nastavime historickou hodnotu  
            textbox.setText("History: "+historyString);  
        } else {  
            textbox.setText("");  
        }  
    }  
});
```



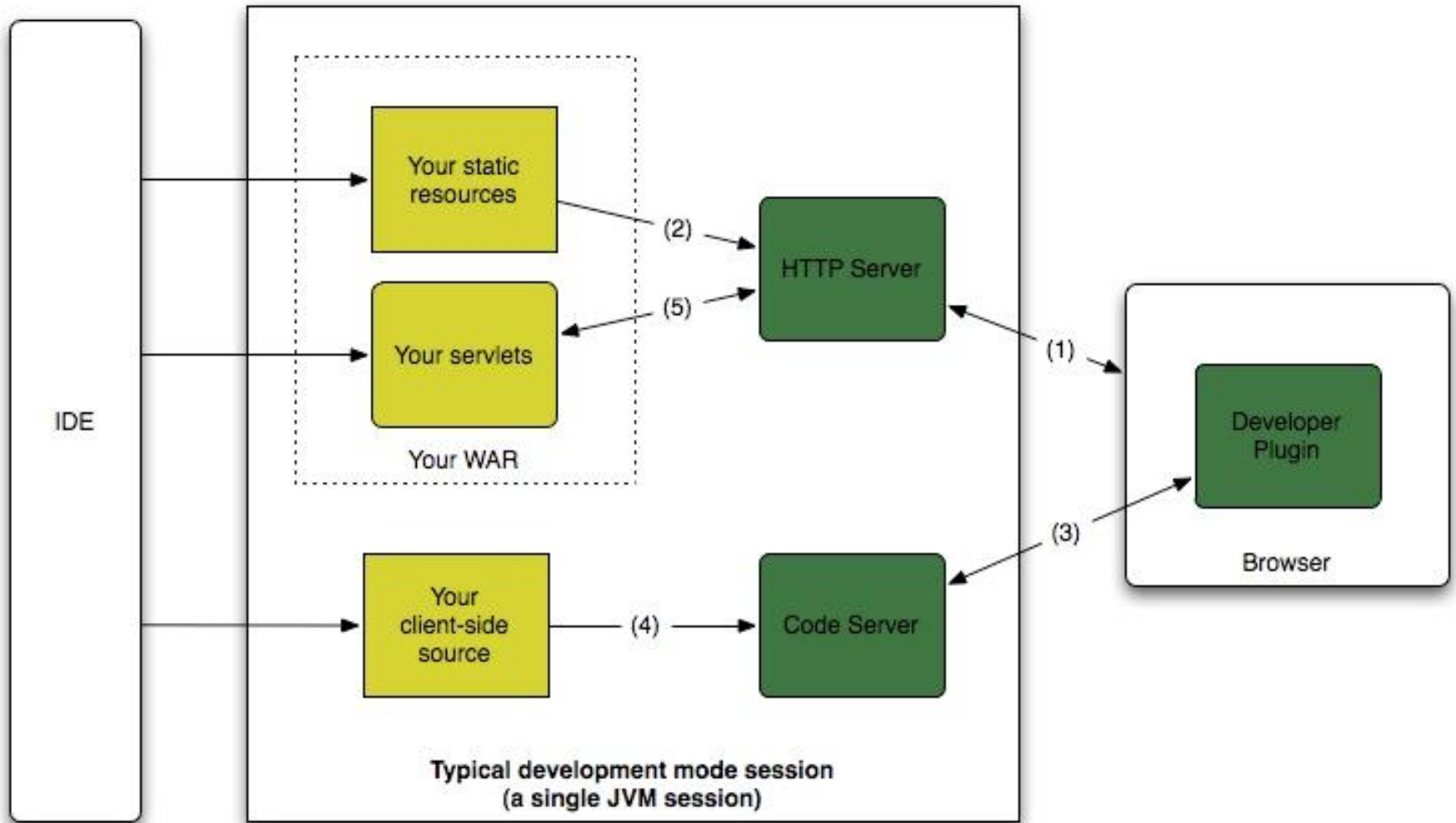
COMPILATION AND DEBUGGING



DCGI



Debugging in Development mode



GWT Development Mode [minimize] [maximize] [close]

Development Mode (FF)

Launch GWT Module

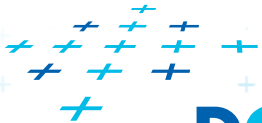
Startup URL: [dropdown arrow]

[Empty content area]



Speed Tracer

- Addon to Chrome
- !!! need for some historical version
- Similar in FF
 - enable-extension-timeline-api



Speed Tracer

Google

← → ↻ 🏠 ☆

🇬🇧 iGoogle 📧 Gmail 🇮🇩 iDNES 🇨🇪 iNed.cz 🌐 X36WWW 🌐 Y36TW1 🌐 Y36TW2 🏠 i2Home 🏠 VITAL 📄 Uživatelé » 📁 Ostatní záložky

🇨🇿 Tato stránka je v jazyce angličtina . Chcete ji přeložit? Přeložit Ne Možnosti ✕

🔴 🔍 📄 total 243.88s zoom 35.77s - 40.82s 🔍 🔍 🔍 http://localhost:8080/Dan ?

100%

Sluggishness (events) 35.77s 36.55s 37.32s 38.09s 38.86s 39.63s 40.40s

Network (resources)

🔍	Started	Duration	Type	Breakdown by Time
	@35.77s	23ms	DOM (mouseover)	100.0% JavaScript...
	@35.79s	21ms	DOM (mousemove)	100.0% JavaScript...
	@35.82s	21ms	DOM (mousemove)	95.5% JavaScript ... 4.5% Dom Event
	@35.84s	20ms	DOM (mousemove)	100.0% JavaScript...
	@35.86s	21ms	DOM (mousemove)	100.0% JavaScript...
	@35.88s	20ms	DOM (mousemove)	100.0% JavaScript...
	@36.07s	27ms	DOM (mousedown)	100.0% JavaScript...
	@36.10s	3ms	Style Recalculation	100.0% Style Rec...
	@36.10s	20ms	DOM (mousemove)	100.0% JavaScript...
Hiding 1 events (0ms)				
	@36.12s	21ms	DOM (mouseup)	100.0% JavaScript...
	@36.15s	34ms	DOM (click)	100.0% JavaScript...
Hiding 1 events (0ms)				
	@36.18s	22ms	DOM (mousemove)	100.0% JavaScript...
	@36.71s	38ms	DOM (mousemove)	100.0% JavaScript...
	@37.73s	32ms	DOM (mousemove)	96.9% JavaScript ... 3.1% Dom Event
	@37.77s	29ms	DOM (mousemove)	96.7% JavaScript ... 3.3% Dom Event
	@37.81s	28ms	DOM (mousemove)	100.0% JavaScript...



Speed Tracer

Google

← → ↻ 🏠 ☆

iGoogle Gmail iDNES iHNed.cz X36WWW Y36TW1 Y36TW2 i2Home VITAL Uživatelé » Ostatní záložky

Tato stránka je v jazyce angličtina . Chcete ji přeložit? Přeložit Ne Možnosti ✕

total 318.32s zoom 35.77s - 40.82s http://localhost:8080/Dan ?

4 requests

Sluggishness (events)

Network (resources)

35.77s 36.55s 37.32s 38.09s 38.86s 39.63s 40.40s

< HTML > damaservice http://localhost:8080/Dam...

Summary

URL	http://localhost:8080/Dama2/cz.cvut.fel.dama/damaservice
From Cache	false
Method	POST
Http Status	200
Mime-type	application/json
Total Bytes	166 bytes
Request Timing	@39816ms for 522ms
Response Timing	@40338ms for 186ms
Total Timing	@39816ms for 709ms

Request Headers

Origin	http://localhost:8080
X-GWT-Module-Base	http://localhost:8080/Dama2/cz.cvut.fel.dama/
User-Agent	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; AppleWebKit/533.2

