

GWT-Gears

The Browser is the Platform

Didier Girard – girard.d@sfeir.com
Sfeir CTO
Member of OSSGTP

Before starting, some questions

- Who knows javascript ?
- Who is a javascript expert ?
- Who knows java ?

Content

- Ajax : the good, the bad and the coward
- Demonstrations
- GWT for Chief (whatyouwant) Officer
- GWT for developers
- Gears for early adopters



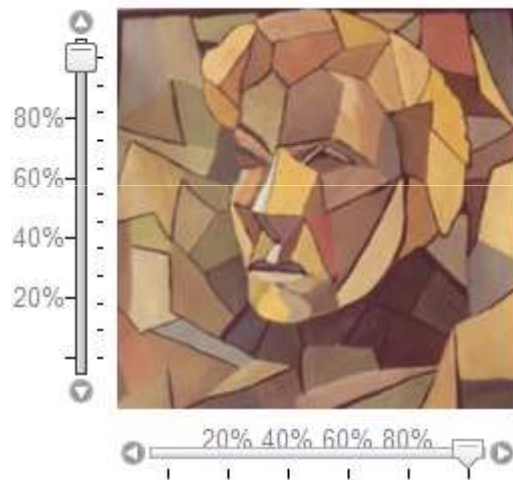
Ajax : Asynchronous updates

- Fresh data without reload



Ajax : Rich Widgets

- Widgets : One more beautiful than the other



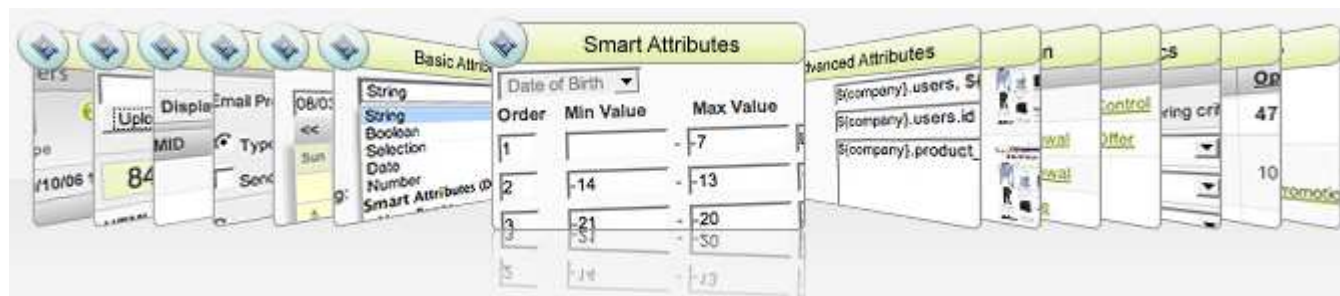
Task	Due Date	Estimate	Rate	Cost
Ext Forms: Field Anchoring				
Integrate 2.8 Forms with 2.0 Layouts	06/24/2007	6 hours	\$150.00	
Implement AnchorLayout	06/25/2007	4 hours	\$150.00	
Add support for multiple types of anchors	06/27/2007	4 hours	\$150.00	
Testing and debugging	06/29/2007	8 hours	\$0.00	
14 Tests	06/29/2007	22 hours	\$112.50	
Ext Grid: Single-level Grouping				
Add required rendering 'hooks' to GridView	07/01/2007	6 hours	\$180.00	
Extend GridView and override rendering functions	07/03/2007	6 hours	\$180.00	
Extend Store with grouping functionality	07/04/2007	4 hours	\$180.00	
Default CSS styling	07/05/2007	2 hours	\$180.00	
Testing and debugging	07/06/2007	6 hours	\$180.00	
15 Tests	07/06/2007	24 hours	\$180.00	
Ext Grid: Summary Rows				
Ext Grid plugin integration	07/05/2007	4 hours	\$125.00	
Summary creation during rendering phase	07/02/2007	4 hours	\$125.00	
Dynamic summary updates in editor grid	07/05/2007	6 hours	\$125.00	
				\$750.00

foo@example.com

Inbox

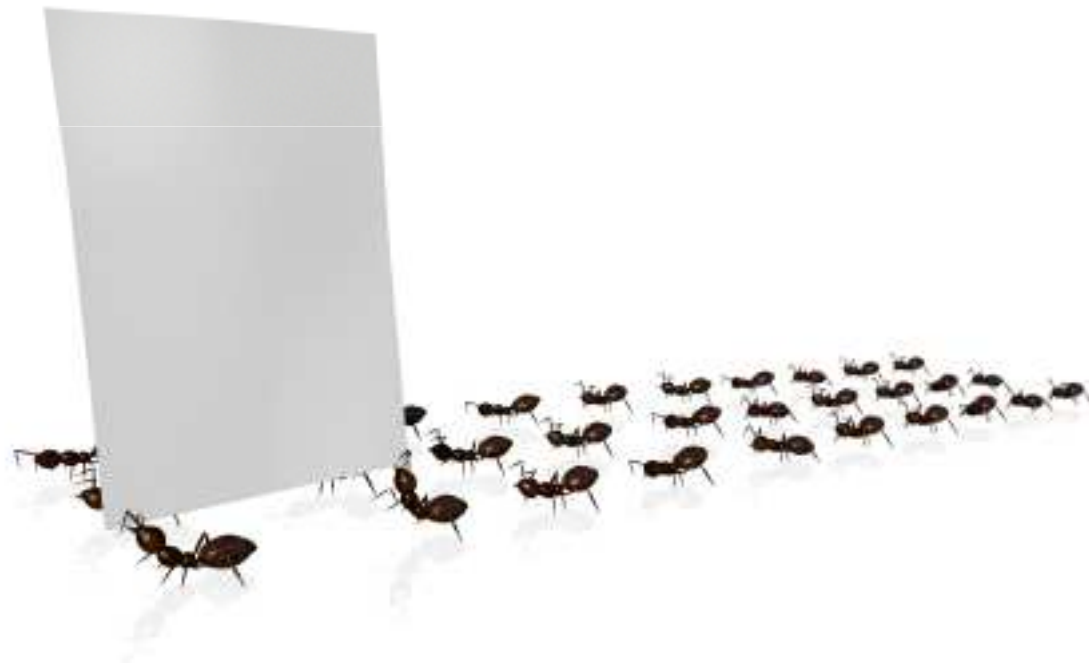
Drafts

Templates



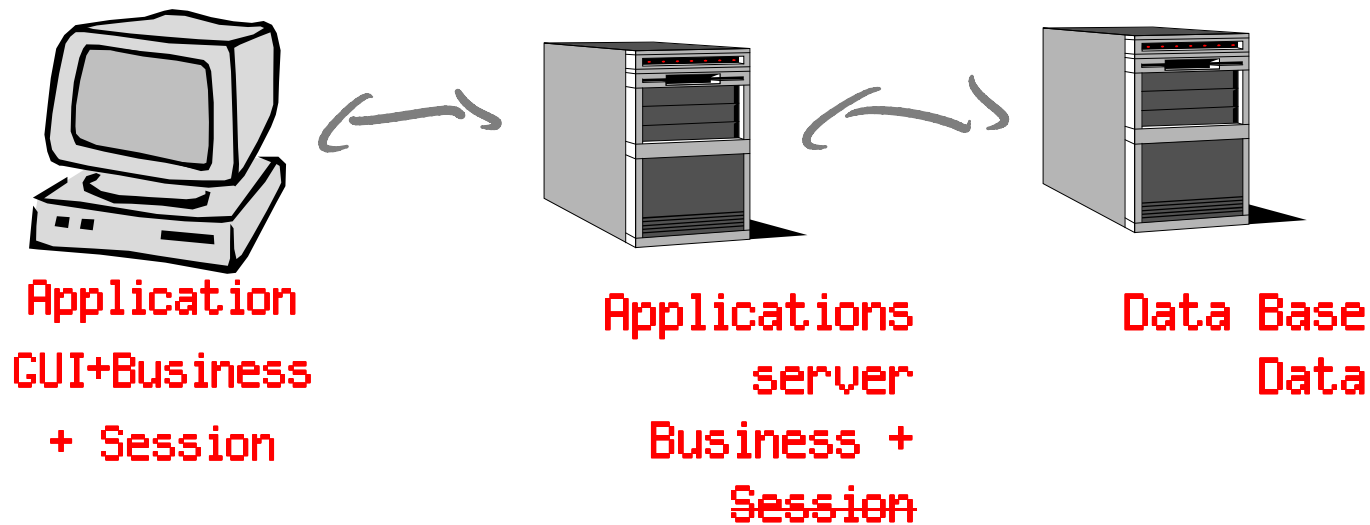
Ajax : Partial update of the screen

- No more « Blank page » effect



Single Page Interface : Better scalability

- Stateless Server
- Stateful client



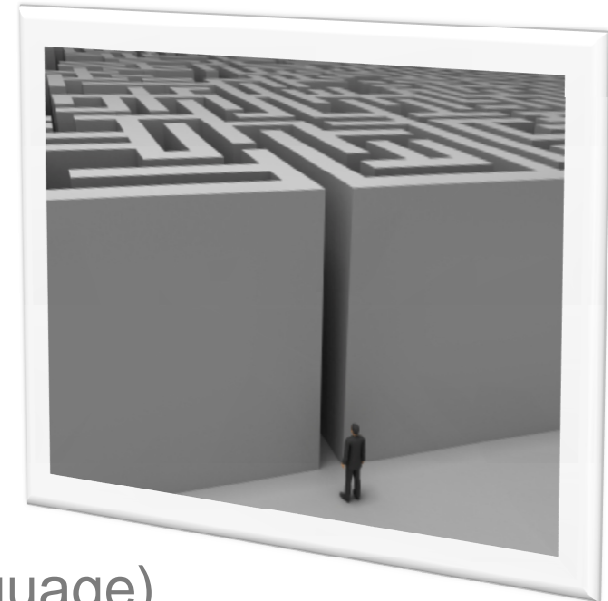
At first, Ajax is really cool

- So wonderful, incredible, marvellous
- So efficient
- Customers love it



Some boring comments about javascript

- ... cross-browser
- ... memory leak
- ... loading latency
- ... « Javascript PhD »
- ... ajax IDE
- ... bugs (javascript is a dynamic language)
- ... cross browser javascript debugger
- ... security
- ...



The solution – First step

- Build the prototype of the application
 - It is really fun
 - Everybody will admire you



The solution – Second step

- Your customer purchases : leave the project :-)



But I'm not
a coward

Do
something
else

But What ?

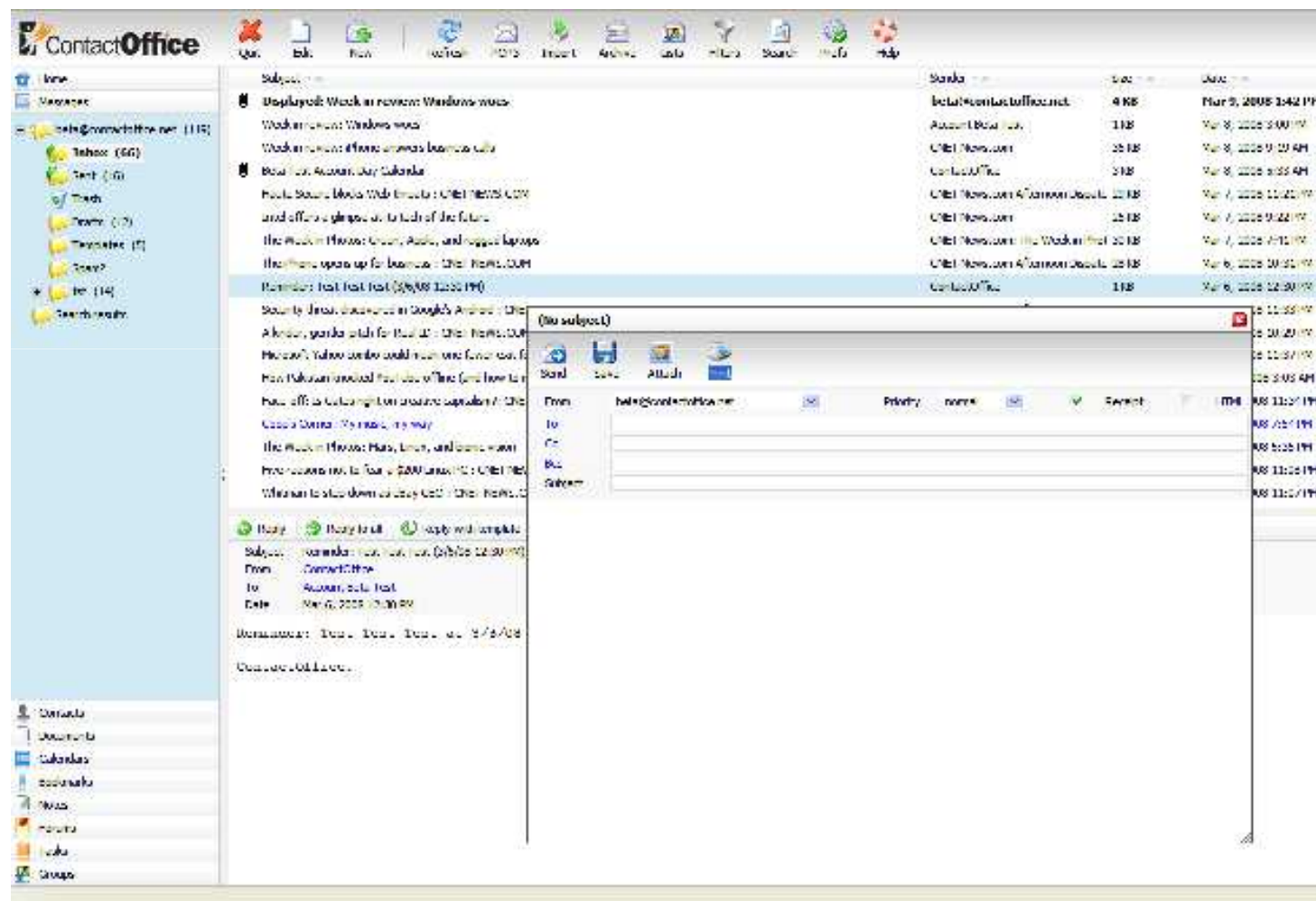
My manifest

- I want a technology that
 - helps to code stable, efficient and cross-browser applications
 - Is a friend to me as a developer
 - Compatible with IDE, support debugging, refactoring, strong typing...
- The base line could be : « Good for the user, good for the developer »

That is
what GWT
is !



Démonstrations



Demonstrations

- <http://stuff.ongwt.com>

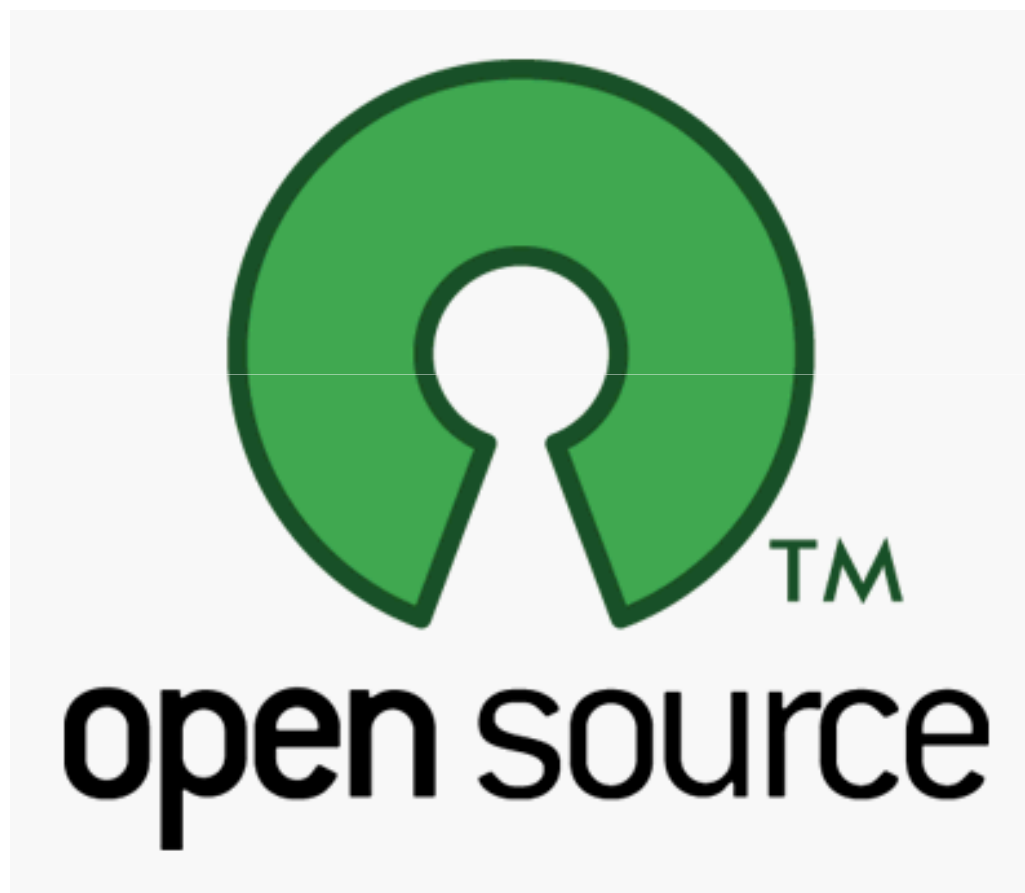


GWT is simple

- Developing a web application with GWT is as simple as developing a Windows application with VB



GWT is cheap



GWT cuts costs

- Building a GWT application is 5 times faster than building a J2EE application



GWT applications are efficient

- Local computation, bandwidth optimization



GWT is offline compatible

- A GWT application doesn't need a server

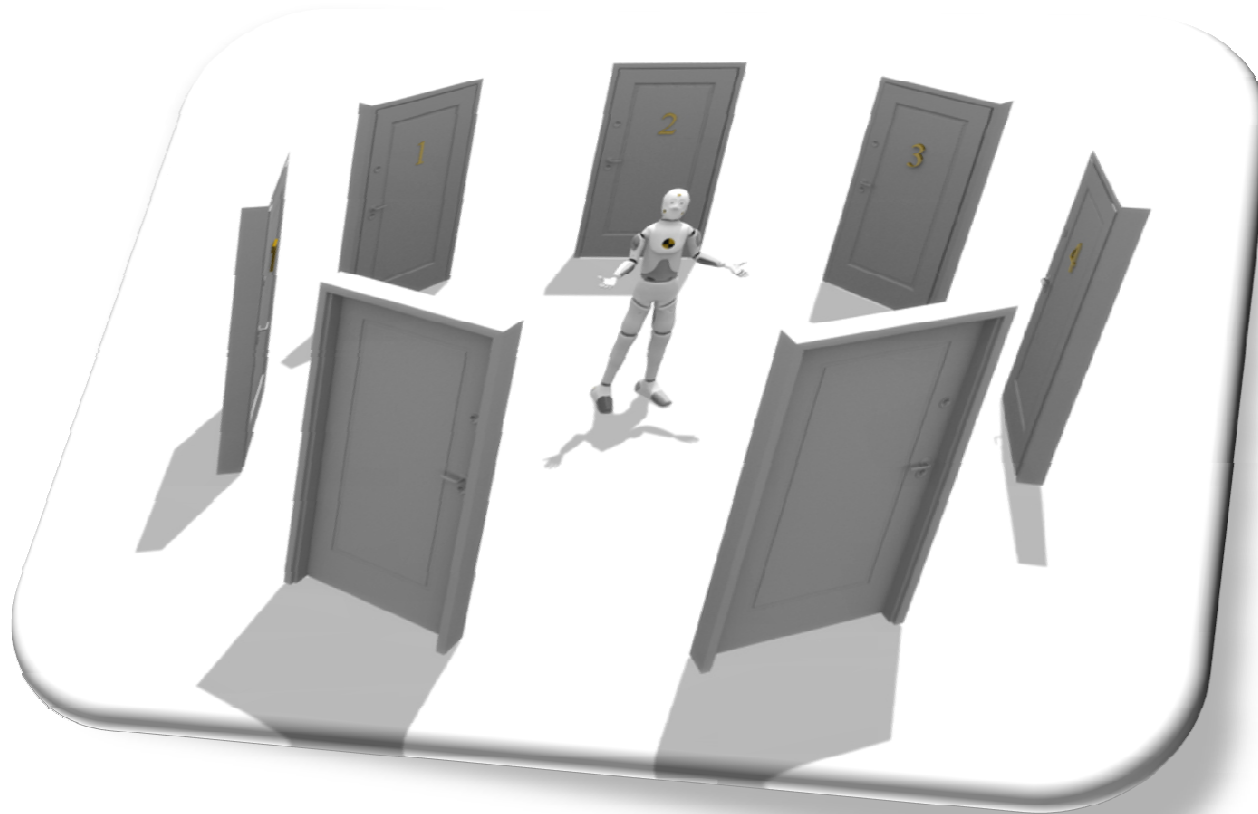


Compatible with mobile



GWT is MDI compatible*

- GWT applications could have windows residing under a single parent window.

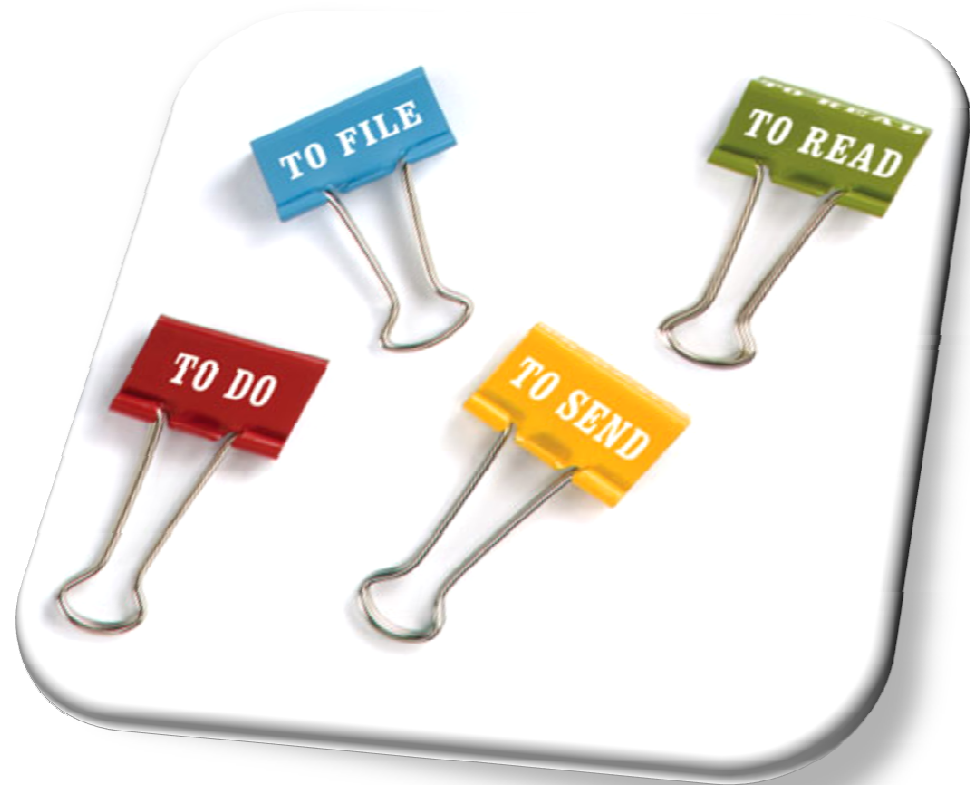


*MDI : Multiple Document Interface

GWT, better ergonomically



Web 1.0



GWT

<http://swissmiss.typepad.com/weblog/2007/11/task-clips.html>



The approach

- Code in java
- Compile the java to Javascript

Hello World

- Demonstration

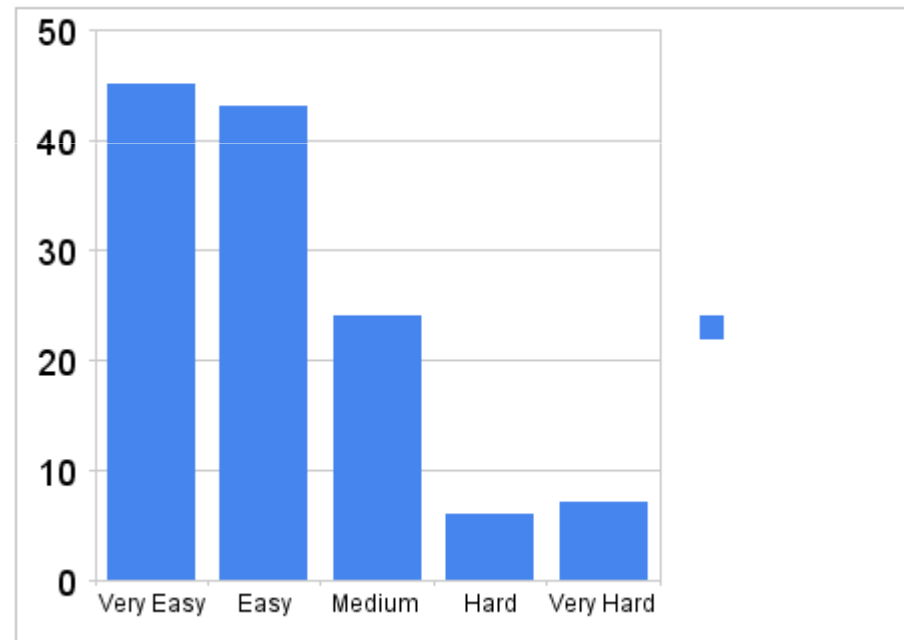
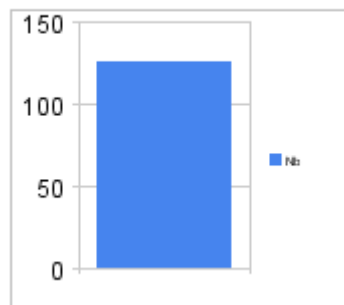
```
public class HelloWorld implements EntryPoint {
    private Button clickMeButton;
    public void onModuleLoad() {
        RootPanel rootPanel = RootPanel.get();

        clickMeButton = new Button();
        rootPanel.add(clickMeButton);
        clickMeButton.setText("Click me!");
        clickMeButton.addClickListener(new ClickListener() {
            public void onClick(Widget sender) {
                Window.alert("Hello, GWT World!");
            }
        });
    }
}
```


How easy do you find GWT to use ?

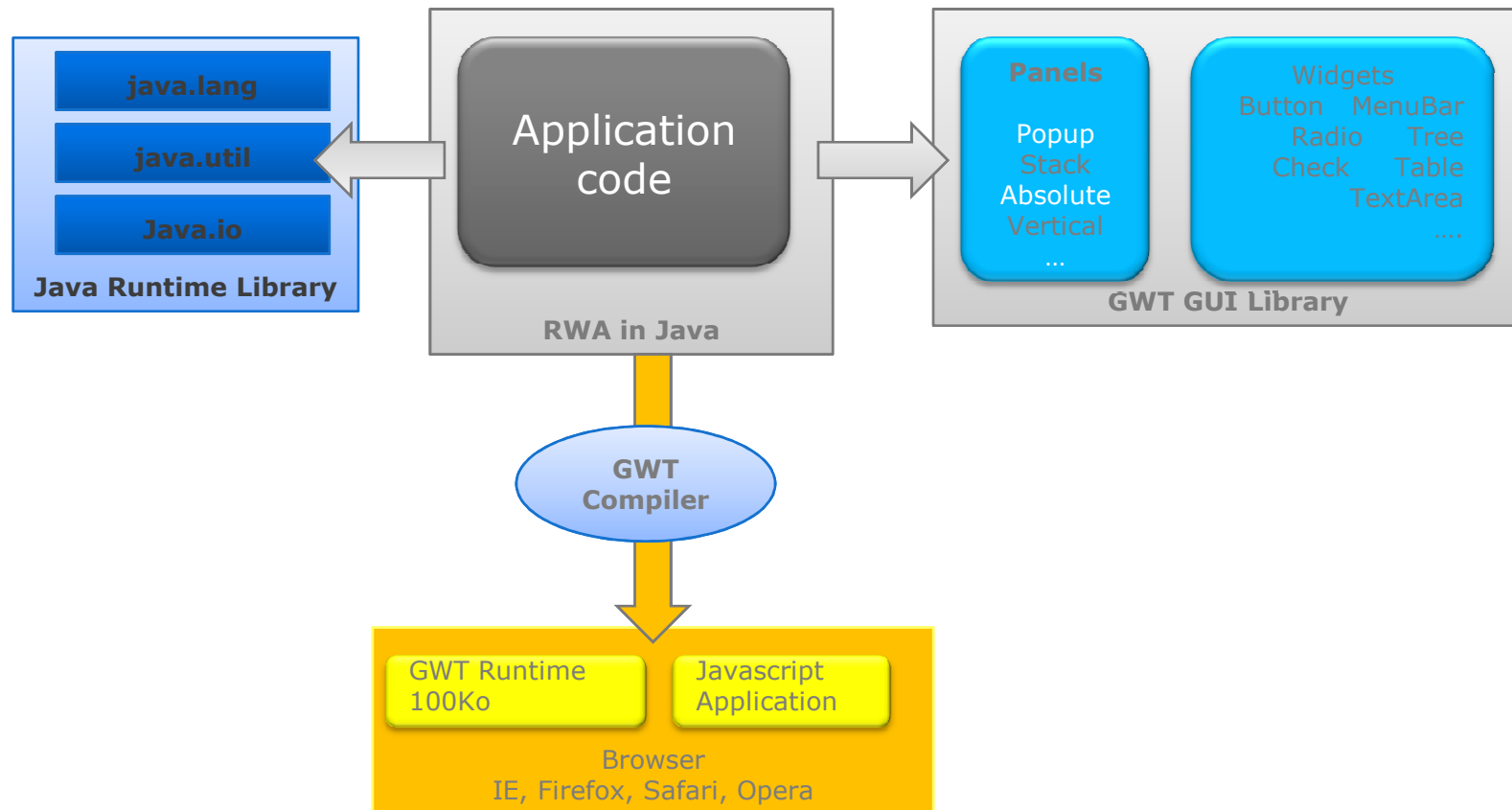
Very Easy, Easy, Medium, Hard, Very Hard ?

Number of votes :



Source : <http://www.ongwt.com/post/2008/02/21/How-easy-do-you-find-GWT-to-use>

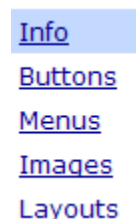
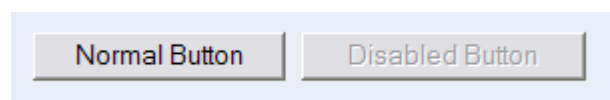
Architecture



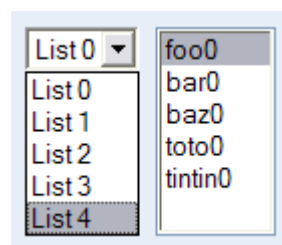


Widgets : « 100% java »

- Demonstration (Kitchen Sink)



sender	email
markboland05	mark@example.com
Hollie Voss	hollie@example.com
boticario	boticario@example.com
Emerson Milton	emerson@example.com
Healy Colette	healy@example.com
Brigitte Cobb	brigitte@example.com
Elba Lockhart	elba@example.com



<http://code.google.com/webtoolkit/examples/kitchensink/>

Third party GWT Widgets

- It is really easy to build its own widget
- Many libraries
 - GWT Ext
 - MyGWT
 - GWT Rialto
 - Tatami
 - GWT Widget
 - GWTaculous
 - ...

History support

- « Single Page Interface » with history support !
- Demonstration (Kitchen Sink)
- Code :

```
History.addHistoryListener(myCommand);
```

RPC Support (JSON/XML-RPC)

- AJAX is a story of asynchronous request
 - With GWT, it is simple !

```
private void getData() {  
  
    DataServiceAsync dataService =  
        (DataServiceAsync) GWT.create( DataService.class );  
  
    ServiceDefTarget endpoint = (ServiceDefTarget) dataService;  
    endpoint.setServiceEntryPoint("/DataService");  
  
    dataService.getData(new AsyncCallback() {  
        public void onSuccess(Object result) {  
            table.setSource(  
                new SimpleDataSource( (Person[]) result ) );  
        }  
  
        public void onFailure(Throwable caught) {  
            Window.alert("Unable to get data from server: "  
                +caught.toString());  
        }  
    });  
}
```

I18N/L10N support

- Simple and powerful

***** Déclaration d'une interface

```
public interface GameStatusMessages extends Messages {  
    String turnsLeft(String username, int numTurns);  
    String currentScore(int numPoints);  
}
```

***** Définition d'un fichier de propriétés (localisées)

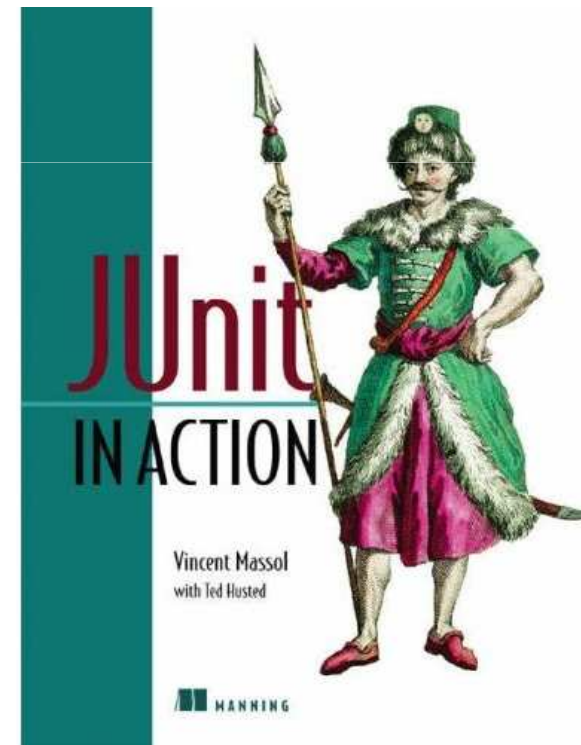
```
turnsLeft = Turns left for player '{0}': {1}  
currentScore = Current score: {0}
```

***** Utilisation

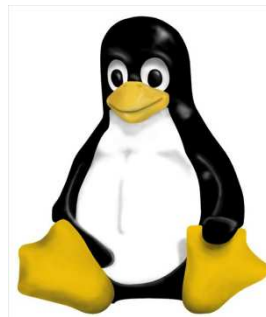
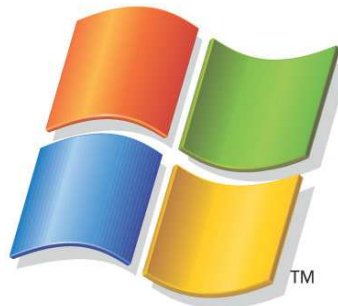
```
GameStatusMessages messages =  
    (GameStatusMessages) GWT.create(GameStatusMessages.class);  
messages.turnsLeft(username, turnsLeft);
```


GWT is compatible with unit testing

- Compatible with Junit
- Compatible with Selenium



Cross browser, cross platform



CSS

- GWT widgets are CSS compatible

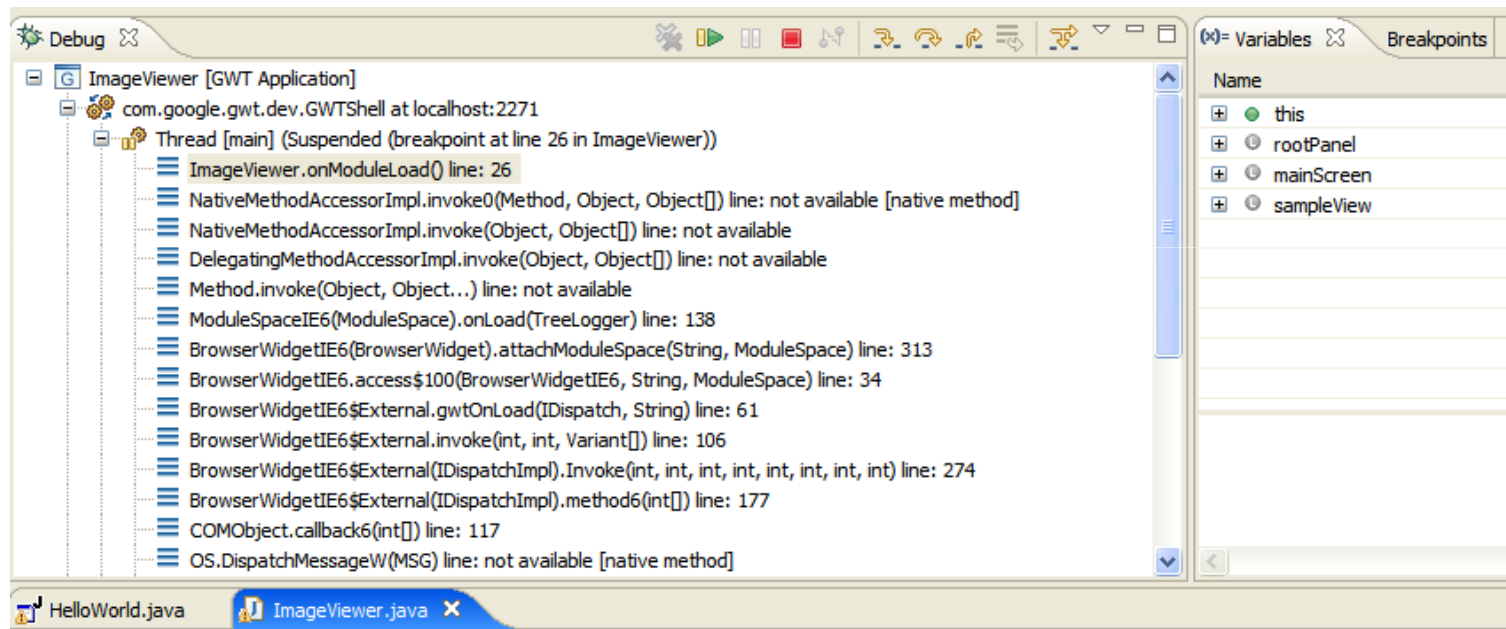


La beauté de la conception CSS



Debugger

- Demonstration



```

SampleView1 sampleView = new SampleView1();
mainScreen.addView(SCREEN1, sampleView);
mainScreen.showView(SCREEN1);
    
```

Javascript integration

- JSNI, a clever API

```
public static native void alert(String msg) /*-{  
    $wnd.alert(msg);  
}-*/;
```

- Many javascript libraries are already wrapped : Dojo, Extjs,...
- Demonstration

Optimization

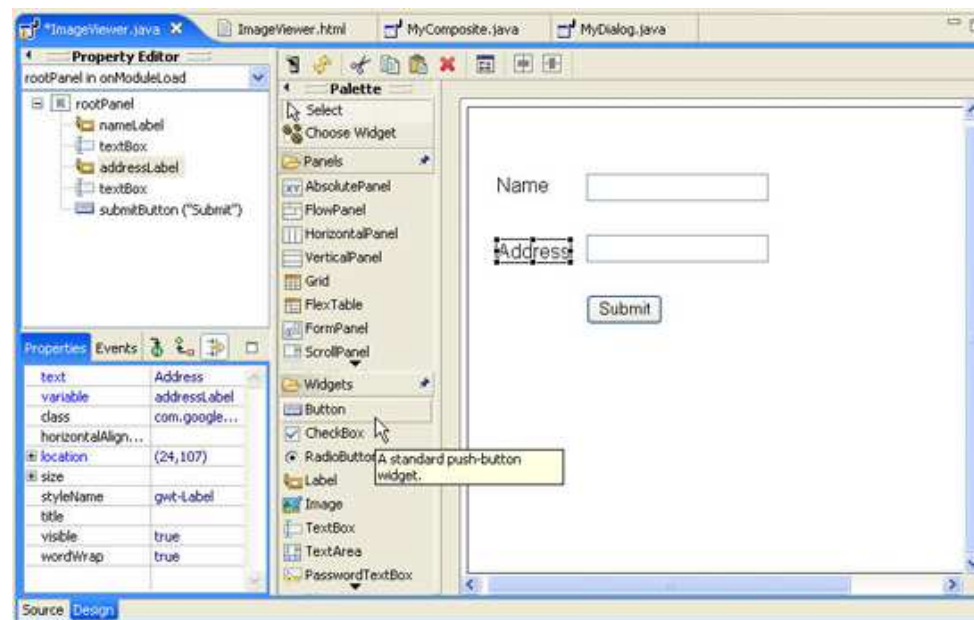
- Compiled javascript is optimized
- Only the necessary javascript is downloaded by the browser
 - IE downloads only javascript for IE, Firefox downloads only javascript for Firefox
 - Unused APIs are removed from the compiled javascript

Good scalability

- GWT server could be stateless
- No more session affinity

Tools

- GWT is supported by all the java IDE : Eclipse, NetBeans, IntelliJ IDEA, JDeveloper,...
- Advertisement : my favorite tool is GWT Designer



GWT 1.4 is fast

- Example : ImageBundle, kind of database for images
 - From 12 requests to 1 request...



```
ClippedImage timeIcon = new ClippedImage("icons.png", 22, 0, 22, 22);
```

<http://roberthanson.blogspot.com/2007/02/gwt-clippedimage-optimizing-image.html>

2008 : GWT 1.5

- The Java 1.5 syntax is fully supported, including generics, enums, nice "for" loops, autoboxing, static imports, annotations, and so on
- The ability to subclass JavaScriptObject for very straightforward JavaScript interop
- Thurs, Mar 6 2008 : GWT 1.5 M1 is available for download

Google Gadget with GWT



Gadget definition

- `<Module>`
`<ModulePrefs title="Hello World"`
`author="Didier Girard"`
`author_email="Didier.Girard@gmail.com">`
`</ModulePrefs>`
`<Content type="html">`
`<![CDATA[`

`]]>`
`</Content>`
`</Module>`

Your application goes here

Gadget definition with GWT : 1 line

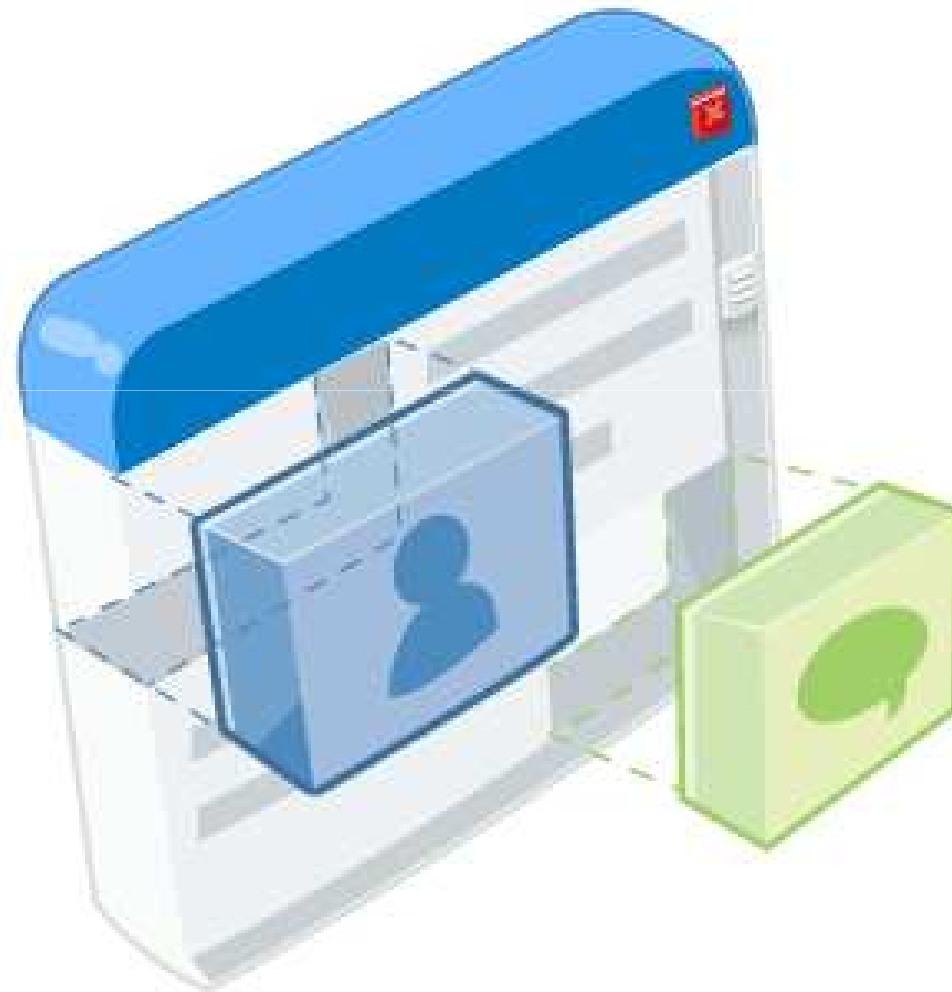
```

• <Module>
  <ModulePrefs title="Hello World"
    author="Didier Girard"
    author_email="Didier.Girard@gmail.com">
    </ModulePrefs>
    <Content type="html">
      <![CDATA[
        <script src="http://.../com.mycompany.project.ImageViewer-xs.nocache.js">
        </script>
      ]]>
    </Content>
  </Module>

```

your application

Opensocial



Who's Using It?



Opensocial Definition with GWT : 2 lines

- ```
<Module>
 <ModulePrefs title="Hello World"
 author="Didier Girard"
 author_email="Didier.Girard@gmail.com">
 <Require feature="opensocial-0.7"/>
 </ModulePrefs>
 <Content type="html">
 <![CDATA[
 <script src="http://.../com.mycompany.project.ImageViewer-xs.nocache.js">
 </script>
]]>
 </Content>
 </Module>
```



# Simple, Efficient and Powerful ?

---





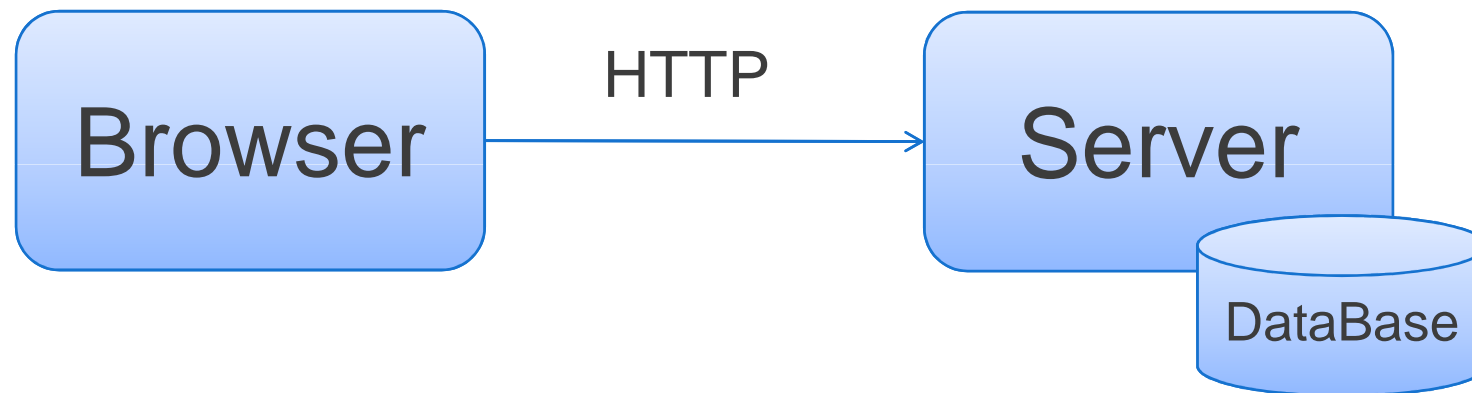
# I need offline applications

---

- Do you ever miss your application
  - Pubs
  - Trains
- Are you sure that your user want its data in the cloud ?

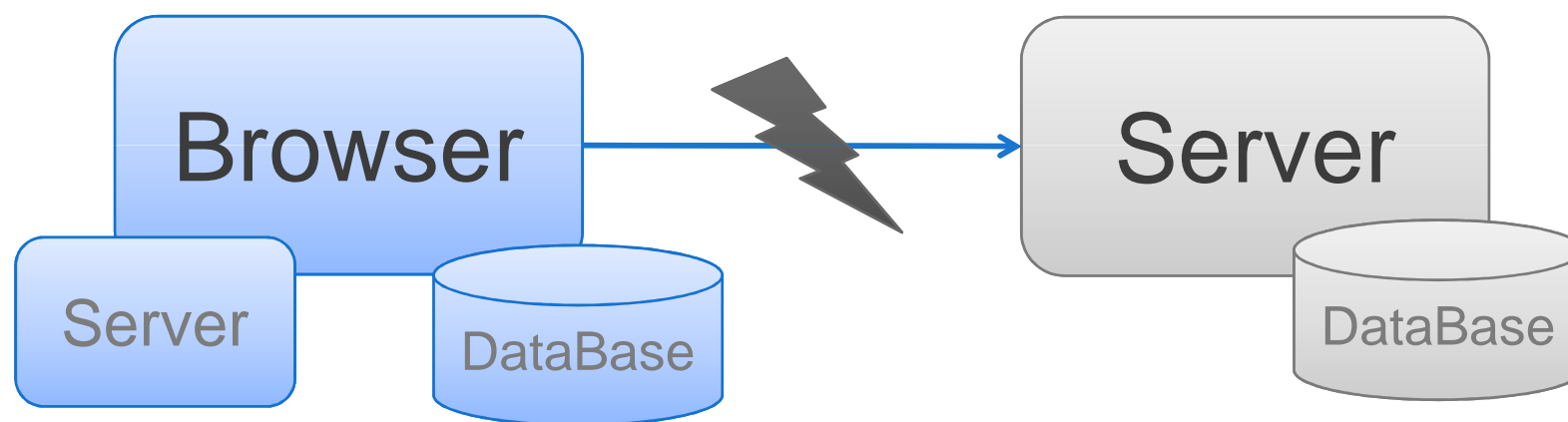
# Classic web architecture

---



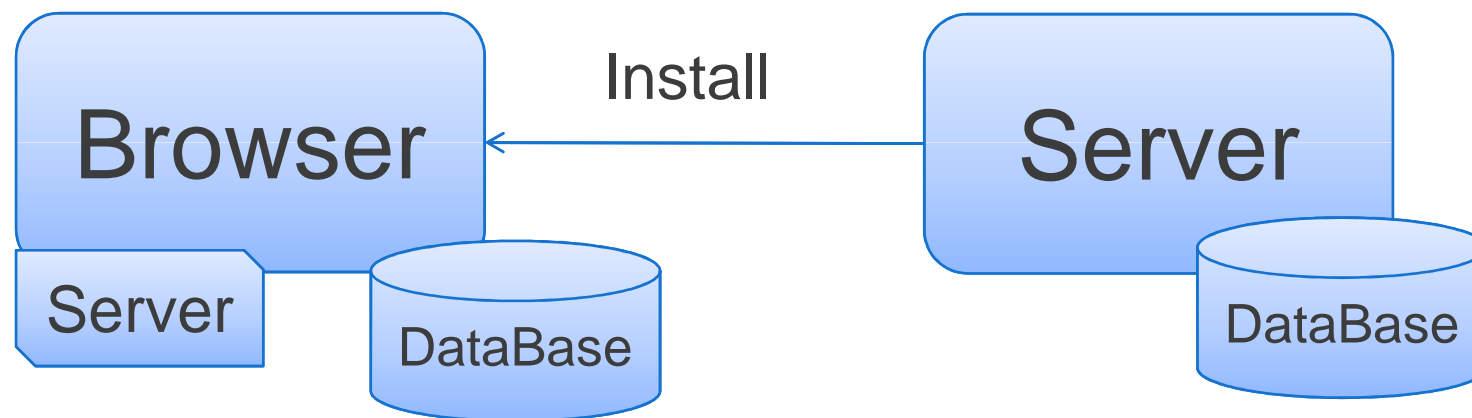
# My needs in offline mode

---



# I need

---



# The solution

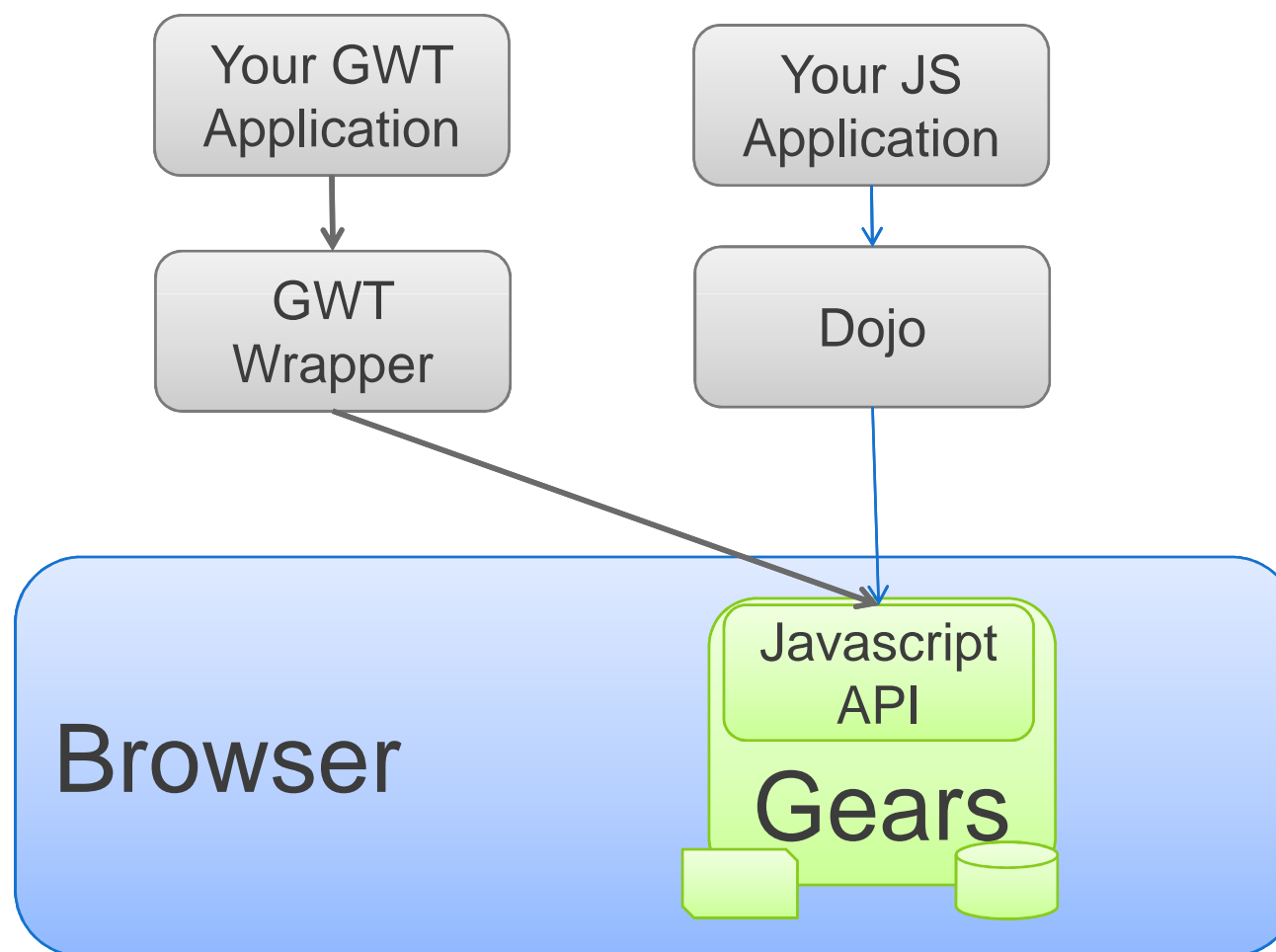
---

- Google Gears
  - Browser Plugin (FireFox, Internet Explorer)
  - Features:
    - LocalServer (“programmable cache”)
    - Database
    - Worker Threads for JavaScript
  - <http://gears.google.com>





# Architecture



# Take Your App Offline

---

1. Server stateless application
2. Manifest file of your app's resources
3. Download resources

# Manifest File

---

```
{
 "betaManifestVersion": 1,
 "version": "Version 1.0",
 "entries": [
 { "url": "index.html" },
 { "url": "style.css" },
 { "url": "resources/logo.gif" },
 { "url": "gears_init.js"},
 { "url": "go_offline.html"},
 { "url": "go_offline.js" }]
}
```

# Load Resources

---

```
LocalServer localServer = new LocalServer();

final ManagedResourceStore managedResourceStore =
localServer.createManagedResourceStore(MANAGED_STORE_NAME);

managedResourceStore.setManifestURL(MANIFEST_URL);
managedResourceStore.checkForUpdate();
```

# Gears Database

---

- Based on SQLite

# As simple as

---

```
private Database m_database = null;

try {
 m_database = new Database("Test");
 ResultSet rs = m_database.execute("create table...");
 rs.close();
} // try
catch (Exception e) {
 // Gears not installed
} // catch
```

# Queries

---

```
String sql = "select id, first_name, last_name from person";
```

```
ResultSet rs = m_database.execute(sql);
```

```
while (rs.isValidRow()) {
```

```
.....
```

```
rs.next();
```

```
}
```

```
rs.close();
```

# Insert

---

```
String args[] = new String[3];
args[0] = Integer.toString(person.getID());
args[1] = person.getFirstName();
args[2] = person.getLastName();
```

```
ResultSet rs = m_database.execute("insert into person
 (id, first_name, last_name) values (?, ?, ?)", args);
rs.close();
```



# Demo

---

# Perspectives of Gears : future APIs

---

- Notification API
- Crypto API
- Factory API updates
- Blob API
- Logging API
- Messaging API
- Location API
- Desktop Shortcut API
- Image Manipulation API



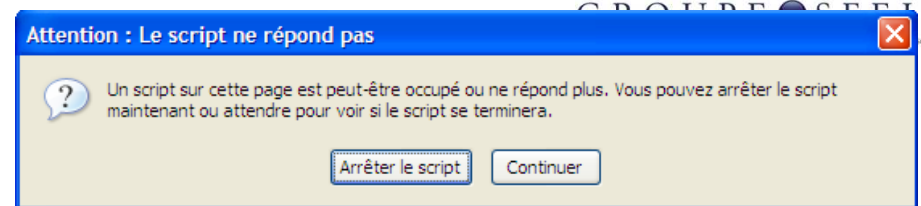
# Think differently!

---

- Rethink application ergonomic.
- Forget browsing
- MDI is possible
- Stateless server, statefull client

# Misc

---



- DOM manipulation is costly
  - Your user can have : « slow script message »
  - -> Use « IncrementalCommand » (DEMO)
- No need to send tones of asynchronous request
- Deal with RPC error by subclassing AsyncCallback
- Forget javascript
- Use Selenium (DEMO)
- Use Maven



# You must used java on the server

---

- No, GWT is server free

# Compiled javascript is impossible to debug

---

- It is possible to change compilation option
- But you never need to it,
- -> Demo



# GWT applications are heavy

---

- Compiled javascript is light and gzippable
  - KitchenSink is around 60Ko !

# GWT is HTML incompatible

---

- Wrong
  - It is possible to include GWT components in a standard HTML page

# GWT applications are applets

---

- Applets, no thanks :-)
  - Java is compiled to javascript !



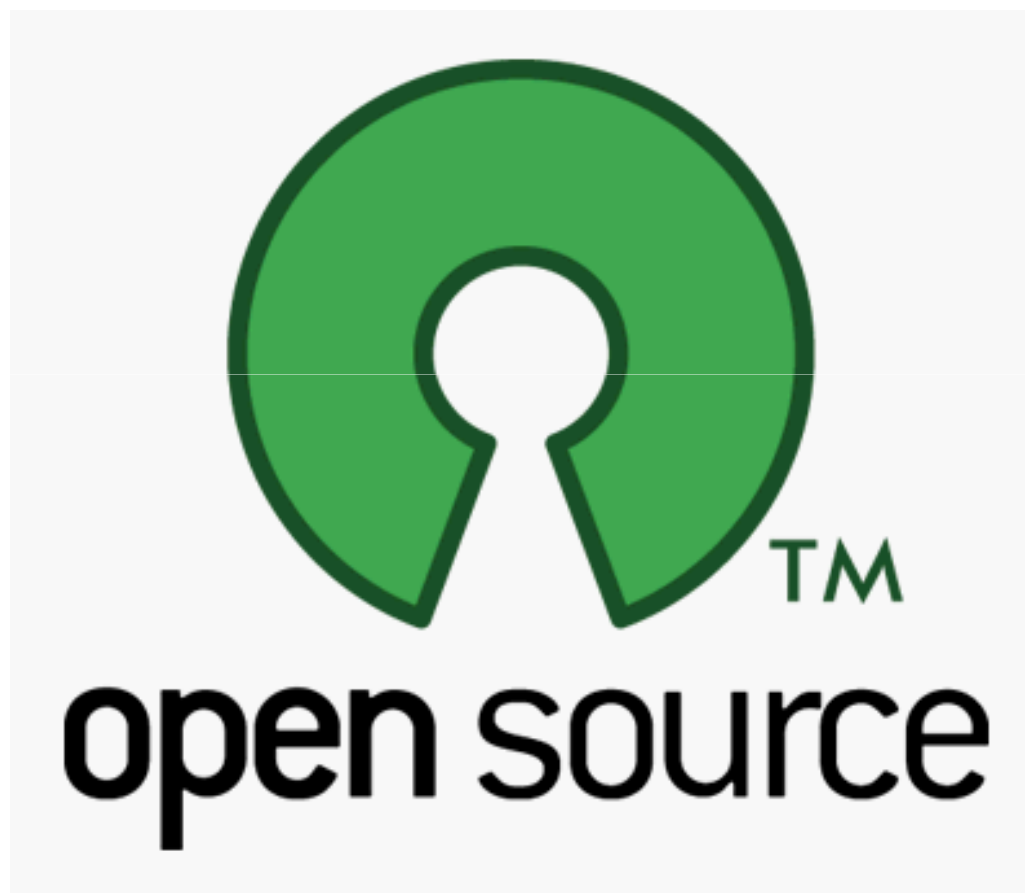
# Who are the challengers

---

- Silverlight
- AIR/Flex
- Firefox with the XUL/Prism stuff

## GWT and Gears

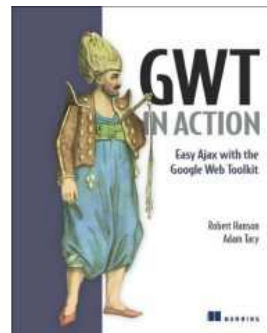
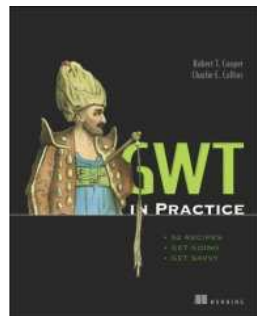
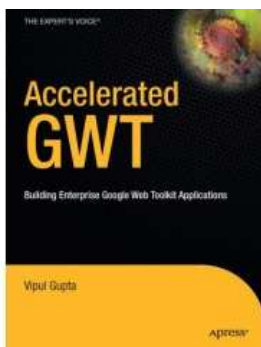
---



# GWT : THE Community

---

- Millions of downloads
- 12000 subscribers to the users forum
- 700 subscribers to the contributors forum
- Books
- Tons of articles
- Tons of tutorials
- Blogs



## Faster and faster

---

- GWT application are efficient
- It is 5 times faster to build a GWT application



## This might surprise you but

---

- This pair of tools is the least disruptive
  - It is based on the most used language : java
  - It doesn't challenge the current web architecture which is mainly based on browsers

# PHP, Struts, JSF, ASP.NET,...

---



## No Future

## Reminder



« GWT + Gears »

The browser is the  
platform !

I thank the GWT Team,  
I thank the community!

# SFEIR

---

- 6 projects with GWT
  - 5000 days of coding
  - It is just fun
  - Clients are very happy with their applications
  - Our stack : GWT/Spring/Hibernate
- Jobs: [girard.d@sfeir.com](mailto:girard.d@sfeir.com)

# Ressources

---

GWT :

<http://code.google.com/webtoolkit/>

Forum :

<http://groups.google.com/group/Google-Web-Toolkit>

GWT Blog :

<http://googlewebtoolkit.blogspot.com/>

onGWT, Tracking news on GWT :

<http://www.ongwt.com/>

GWT and GEARS :

<http://www.ongwt.com/post/2008/01/28/Presentation-%3A-Gears-and-GWT>