

GWT: The Technical Advantage

IndicThreads.com Conference On Java Technology

PUNE, INDIA

25

26

27

Nov. 2008

Presenter: Anirudh Dewani

Company Name: Google

What is GWT?



How it works



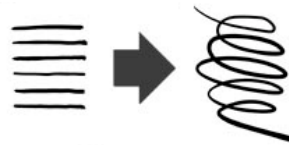
Code against Java UI libraries



How it works

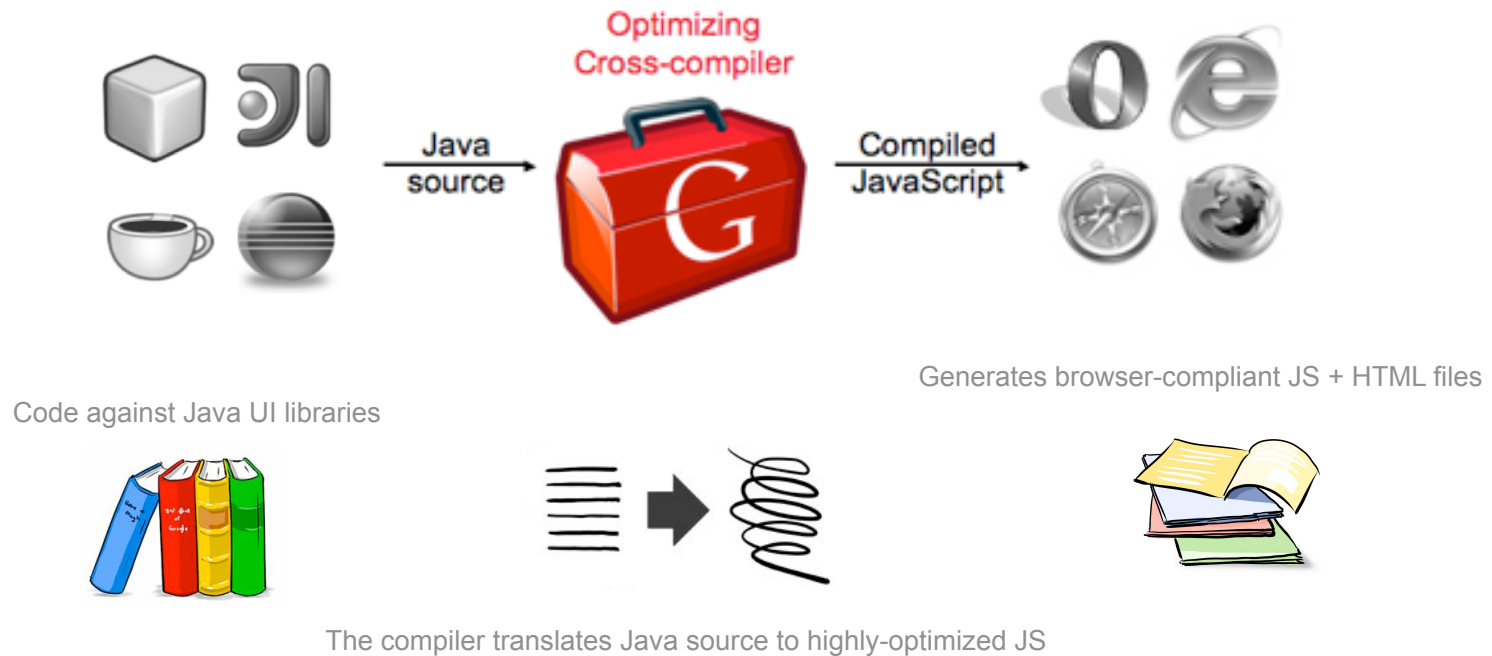


Code against Java UI libraries

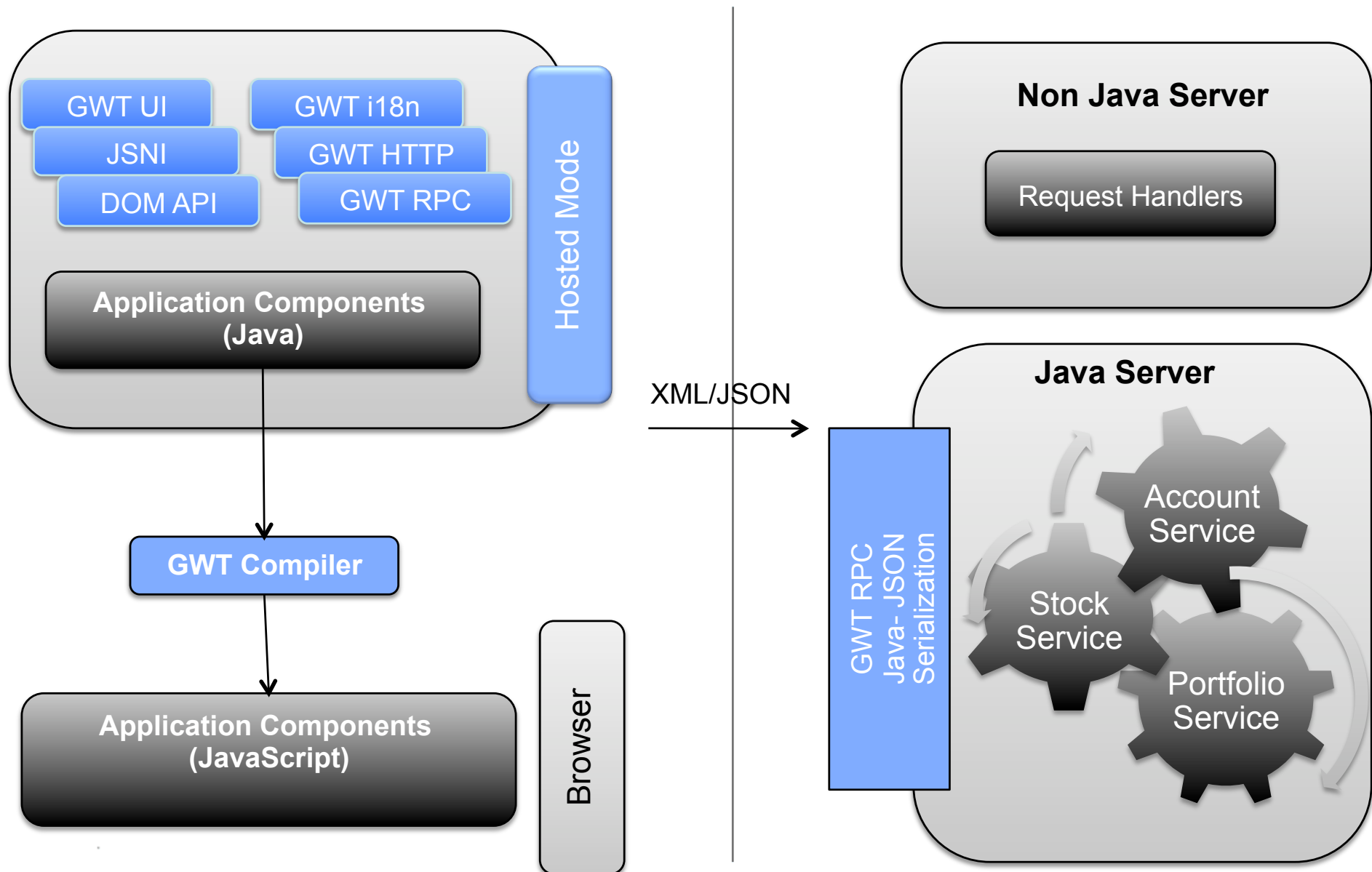


The compiler translates Java source to highly-optimized JS

How it works



GWT – High Level Overview



What are the advantages of this approach?

GWT Advantages – Faster Ajax applications



Faster-than-you-would-write-by-hand code

- The efficient code you wish you could write, but will get slammed by cross-browser issues for trying to run it

E.g.

```
public static void onModuleLoad() {  
    Button b = (new Button()).Button();  
    b.setText("w00t!");  
}
```

– After a few compiler visitors, this becomes:

```
public static final void onModuleLoad() {  
    final Button b = Button.$Button(new Button());  
    DOMImplIE6.$setInnerText(b.element, "w00t!");  
}
```

GWT Advantages – Faster Ajax applications



Faster-than-you-would-write-by-hand code

- After a few more compiler visitors, it becomes:

```
function onModuleLoad() {  
    var b;  
    b = $Button(new Button());  
    $setInnerText(b.element, 'w00t!');  
}
```

GWT Advantages – Faster Ajax applications



Faster-than-you-would-write-by-hand code

- After a few more compiler visitors, it becomes:

```
function onLoad() {  
    var b;  
    b = $Button(new Button());  
    $setInnerText(b.element, 'w00t!');  
}
```

- You could have written this by hand, but:
 - You would have to change it for every other browser
 - You could hide it behind an abstraction, but it would add more virtualization than your users care for.

GWT Advantages – Faster Ajax applications



Free optimizations

- You just keep writing GWT code, let the compiler worry about optimizing it.
- BUT! That doesn't mean that general good programming practices don't apply i.e. Inefficient algorithms, redundant objects etc..
- Also.. CSS complexities, elaborate DOM constructs etc

GWT Advantages – Faster Ajax applications



Deferred binding

- Why give the user more than they asked for?
- Users only download what they need to run your application
- Made possible through the technique of **deferred binding**

GWT Advantages – Faster Ajax applications

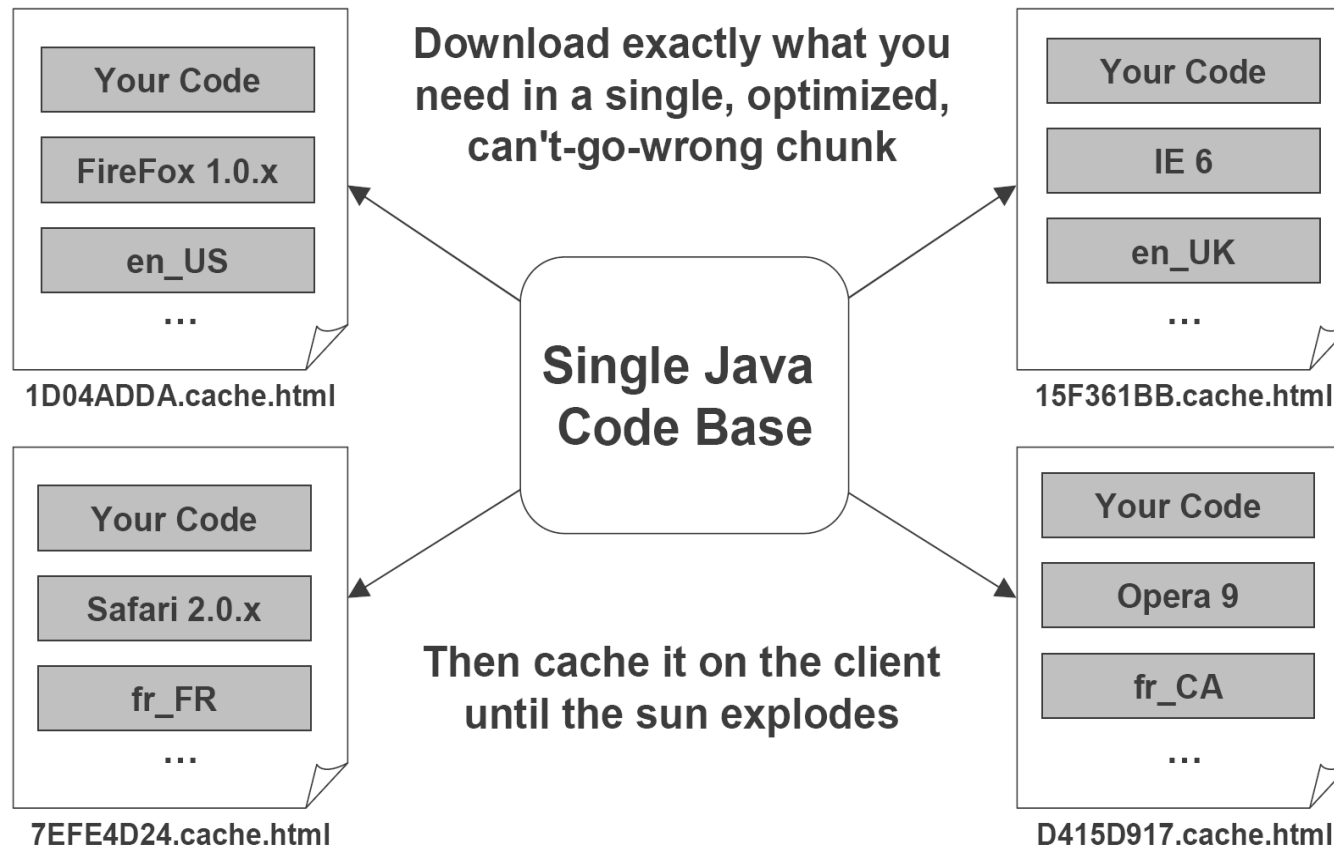


Deferred binding

- A technique that lets the compiler make different bindings for your application at compile-time and choose the right one later
- The application bootstrap process selects the correct binding when loading your application

GWT Advantages – Faster Ajax applications

Deferred binding illustrated



GWT Advantages – Faster Ajax applications



Deferred binding in code

RequestBuilder class

```
private static final HTTPRequestImpl httpRequest =  
(HTTPRequestImpl) GWT.create(HTTPRequestImpl.class);
```

HTTPRequest.gwt.xml

```
<!-- Fall through to this rule is the browser isn't IE -->  
<replace-with class="com.google.gwt.user.client.impl.HTTPRequestImpl">  
  <when-type-is class="com.google.gwt.user.client.impl.HTTPRequestImpl"/>  
</replace-with>  
  
<!-- IE differs slightly in how XmlHttpRequest gets instantiated -->  
<replace-with class="com.google.gwt.user.client.impl.HTTPRequestImplIE6">  
  <when-type-is class="com.google.gwt.user.client.impl.HTTPRequestImpl"/>  
  <when-property-is name="user.agent" value="ie6"/>  
</replace-with>
```

GWT Advantages – Skip the browser quirks



Example

- You code to an abstraction of a given widget

```
final PopupPanel popup = new PopupPanel();  
popup.center(); //center and show the popup
```

GWT Advantages – Skip the browser quirks



Example

- You code to an abstraction of a given widget

```
final PopupPanel popup = new PopupPanel();  
popup.center(); //center and show the popup
```

- The GWT compiler takes care of subbing in the right implementation

PopupImpl class

```
public void onShow(Element popup) {  
}
```

PopupImplIE6 class

```
public native void onShow(Element popup) /*- {  
    var frame = $doc.createElement('iframe');  
  
    // Setting a src prevents mixed-content warnings.  
    frame.src = "javascript:''";
```

GWT Advantages – No more memory leaks



Preventing memory leaks

- Provided you only code in GWT
- Chances are, you may need to write a small bit of JavaScript Native Interface (JSNI) or interoperate with JavaScript code
- In those cases, you can prevent memory leaks by being careful
 - See Joel Webber's article on [“DOM events, memory leaks, and you”](#)
- In every other case, GWT has got your back

GWT Advantages – History support

History support for your GWT applications

- GWT offers History support (RSH implementation)

E.g.

```
tabPanel.add(new HTML("<h1>Page 1 Content</h1>"), " Page 1 ");
tabPanel.add(new HTML("<h1>Page 2 Content</h1>"), " Page 2 ");
tabPanel.addTabListener(new TabListener() {
    @Override
    public void onTabSelected(SourcesTabEvents sender, int tabIndex) {
        // Push an item onto the history stack
        History.newItem("page" + tabIndex);
    }
});
```

```
History.addHistoryListener(new HistoryListener() {
    public void onHistoryChanged(String historyToken) {
        if(tokenIsValid(historyToken)) {
            tabPanel.selectTab(getTabIndex(historyToken));
        }
    }
});
```

GWT Advantages – Code reuse

Code reusability through design patterns

- Gang of Four: Observer, Mediator, Strategy, ...

GWT Advantages – Code reuse

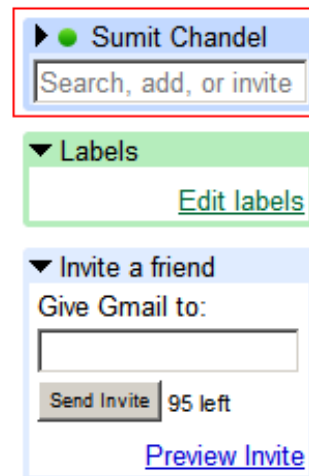
Code reusability through design patterns

- Gang of Four: Observer, Mediator, Strategy, ...
- **Composite pattern in action**

GWT Advantages – Code reuse

Code reusability using the Composite pattern

- UI component being designed



The screenshot shows a vertical stack of UI components. At the top is a blue header bar with a green dot and the name 'Sumit Chandel'. Below it is a search bar with the placeholder text 'Search, add, or invite'. The next section is a green bar with a dropdown arrow and the text 'Labels', containing an 'Edit labels' link. The final section is a blue bar with a dropdown arrow and the text 'Invite a friend', containing a 'Give Gmail to:' label, an empty text input field, a 'Send Invite' button with '95 left' next to it, and a 'Preview Invite' link.

- A set of specialized dialog boxes

GWT Advantages – Code reuse

Code reusability with the Composite pattern

- Start with the generic composite

```
public abstract class GmailDisclosurePanel extends Composite {
    DisclosurePanel disclosurePanel;

    public GmailDisclosurePanel() {
        disclosurePanel = getDisclosurePanel();
        initWidget(disclosurePanel);
    }

    protected abstract DisclosurePanel getDisclosurePanel();
}
```

GWT Advantages – Code reuse

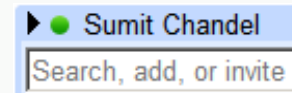
Code reusability with the Composite pattern

- Create the specialized subclasses: Contact

```
public class ContactDisclosurePanel extends
GmailDisclosurePanel {

    public ContactDisclosurePanel() {
        ContactList contactList = new ContactList();
        disclosurePanel.add(contactList);
    }

    @Override
    protected DisclosurePanel getDisclosurePanel() {
        VerticalPanel contactPanel = new VerticalPanel();
        contactPanel.add(new Label(Profile.getName()));
        TextBox textBox = new TextBox();
        textBox.setText("Search, add or invite");
        contactPanel.add(textBox);
        return new DisclosurePanel(contactPanel);
    }
}
```



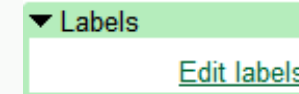
GWT Advantages – Code reuse

Code reusability with the Composite pattern

- Create the specialized subclasses: Label

```
public class LabelDisclosurePanel extends GmailDisclosurePanel {
    public LabelDisclosurePanel() {
        Hyperlink editLabelLink =
            new Hyperlink("Edit labels", "editLabelLink");
        disclosurePanel.add(editLabelLink);
    }

    @Override
    protected DisclosurePanel getDisclosurePanel() {
        return new DisclosurePanel("Labels");
    }
}
```



GWT Advantages – Code reuse

Code reusability with the Composite pattern

- Composites are good because:
 - They let you easily reuse components
 - Let you control access to underlying widgets
- In fact... we use Composite in the GWT UI Libraries!

```
public class TabPanel extends Composite implements TabListener,  
SourcesTabEvents, HasWidgets, HasAnimation, IndexedPanel {
```

```
public class CaptionPanel extends Composite implements HasWidgets {
```

```
public final class DisclosurePanel extends Composite implements  
FiresDisclosureEvents, HasWidgets, HasAnimation {
```

GWT Advantages – Faster development



Faster development with IDEs and code support

- Most developers are familiar with code refactoring and the tools that make it easier
- But... in case you didn't know, here's a demo of using code refactoring (in Eclipse) for a GWT Ajax application

GWT Advantages – Faster development

Faster development with IDEs and code support

- You can also thoroughly test your Ajax application using a combination of:
 - Standard JUnit TestCase
 - GWTTestCase
 - Selenium test

GWT Advantages – Debugging in bytecode



Debugging with hosted mode

- We already saw this earlier... but there's more
- Making hosted mode even more powerful by taking it out-of-process

What are the advantages of this approach?

- Optimized, high performance Ajax applications
- As a developer, you don't have to worry about:
 - Browser quirk headaches
 - Memory leaks
 - History support
- Code reuse through design patterns
- Faster development using IDEs and other Java tools
- Debugging in bytecode

What's new in GWT 1.5

What's new in GWT 1.5?

- GWT 1.5 released August 28th, 2008, includes:
 - Java 5 support
 - Easier interop with JavaScript using JSO overlays
 - Enhanced DOM class for full specification compliance
 - Better application performance

Learn more
code.google.com/webtoolkit

Q&A