



Papick Garcia Taboada (pgt/ adminSight)

**GWT**

JSF done right?

# Session topics

- **Web 2.0, Ajax**
- GWT – What is it?
- Java EE and the Web
- GWT and Java EE
- JSF done right?

# Web 2.0

- Hard to define – what is the web 2.0?
  - Set of public web applications?
    - Survivors of the .com crash?
  - Set of Buzzwords?
  - Set of technologies used?

# Web 2.0 by apps

- DoubleClick → Google AdSense
- Ofoto (Kodakgallery) → Flickr
- „Akamai → BitTorrent
- mp3.com → Napster (iTunes?)
- Britannica Online → Wikipedia
- personal websites → blogging
- Evite → Upcoming.org and EVDB
- domain name speculation → search engine optimization
- page views → cost per click
- screen scraping → web services
- publishing → participation
- content management systems → wikis
- directories (“taxonomy“) → tagging (“folksonomy“)
- stickiness → syndication

# Web 2.0 by buzzwords

- RSS
- Atom
- Trackbacks
- Pingbacks
- Social bookmarking
- Podcasting (audio, video)
- Blogs, blogosphere
- RIAs – Rich Internet Applications
- MASHUPS!

# Web 2.0 by technologies

- Builds on known DHTML technologies
  - HTML
  - Javascript
  - CSS
- Less plugin based!
  - no Java applets
  - no Flash applets
  - etc...

# Web 2.0 what we see

- Rich user experience
  - by great ideas
  - by great user interface
  - by good services
  - by low pricing

# The web **THEN** and now

- Browsers where dumb html terminals
  - Page roundtrips
  - Latency
  - High amount of data transferred over and over again
  - High server side resource consumption

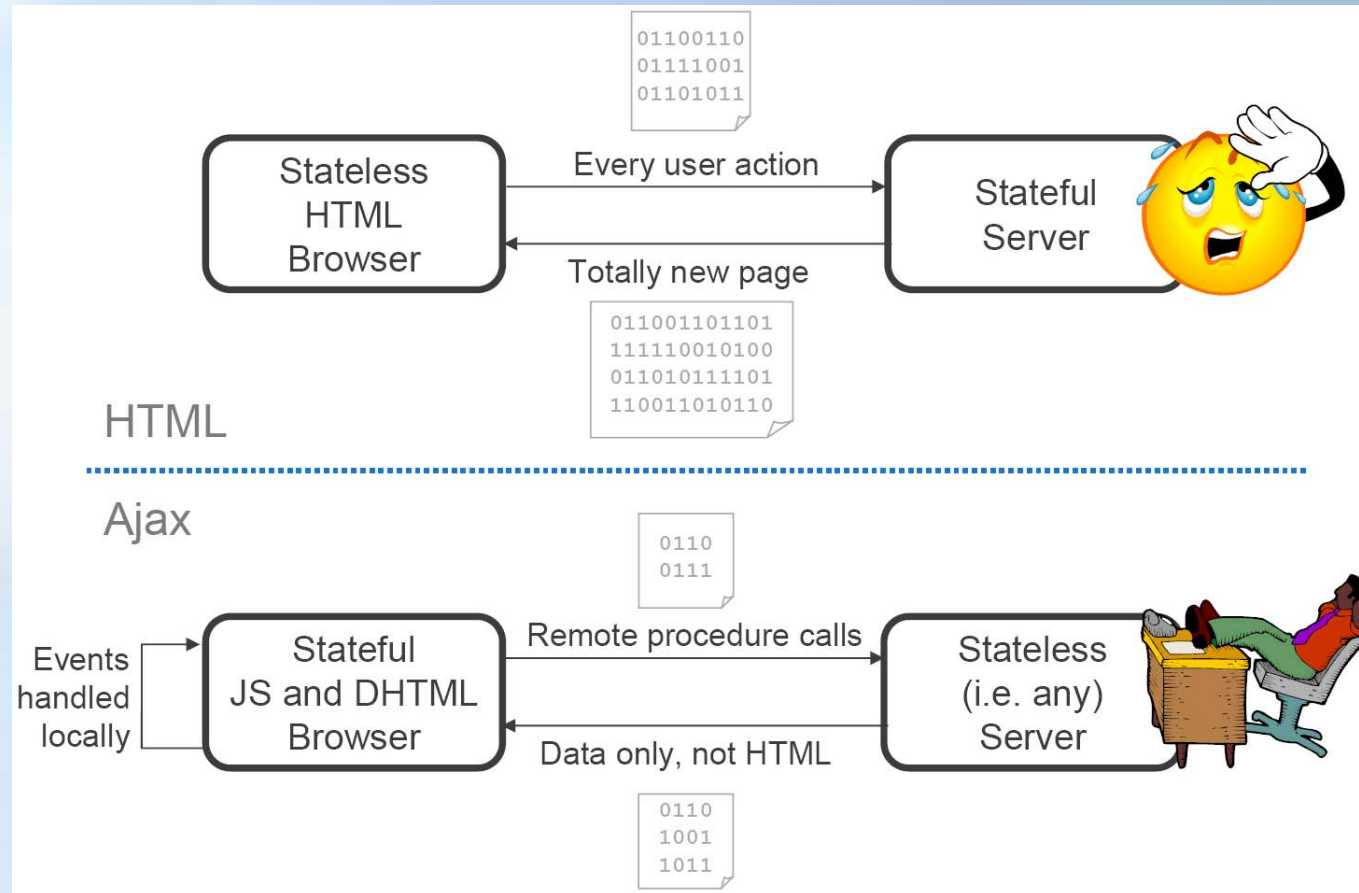
# The web then and **NOW**

- With Ajax
  - Background, asynchronous communication
- With DHTML
  - Application runtime platform
- Easy, really VERY \*easy\* deployment
- Awesome huge user basis

# Web 2.0 – the new integration platform

- Mashups
  - More than eye candy
  - Integration happens on the client
- Browser as runtime engine
- JS as programming language (outch)
- Google (and others) as service providers

# The Ajax architectural shift



# The Ajax slippery slope

- Try “a little bit of script”
- Every body loves it! More!!!
  - Errr - Do we know Javascript?
  - Does it work with...
    - Wait – this is hard

# Fear browsers with all of mind, body and spirit

When in danger,  
when in doubt, run in  
circles cry and shout



# Session topics

- Web 2.0, Ajax
- **GWT – What is it?**
- Java EE and the Web
- GWT and Java EE
- JSF done right?

# GWT – Google Web Toolkit

- Google Web Toolkit (GWT) is an **open source** Java development **framework** that lets you **escape the matrix of technologies** that make writing AJAX applications so difficult and error prone. With GWT, you can **develop and debug** AJAX applications in the Java language using the **Java development tools** of your choice. When you deploy your application to production, the GWT compiler **translates** your Java application to **browser-compliant JavaScript** and HTML.

# GWT development cycle #1

- Use your favorite Java IDE to write and debug an application in the Java language, using as many (or as few) **GWT libraries** as you find useful.

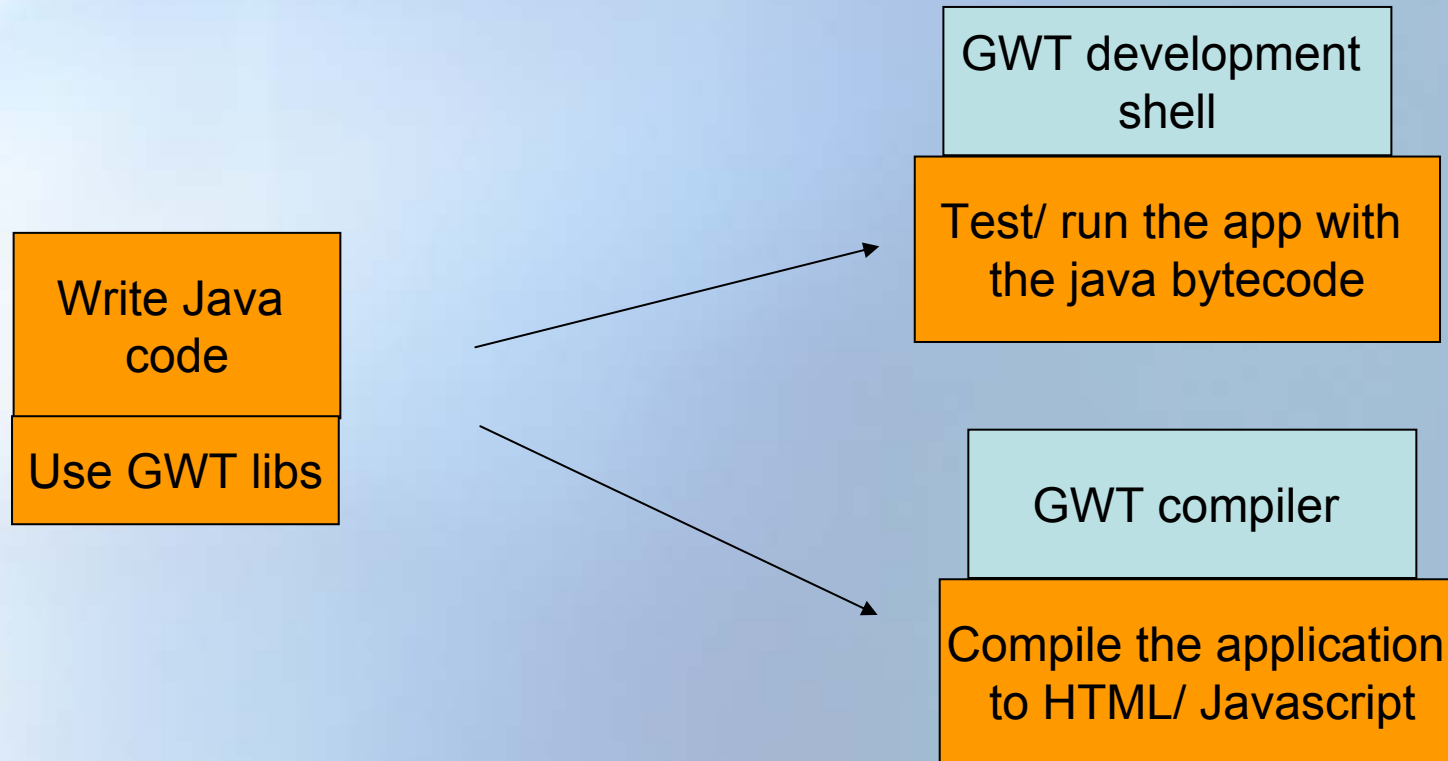
# GWT development cycle #2

- Use GWT's **Java-to-JavaScript compiler** to distill your application into a set of JavaScript and HTML files that you can serve with any web server.

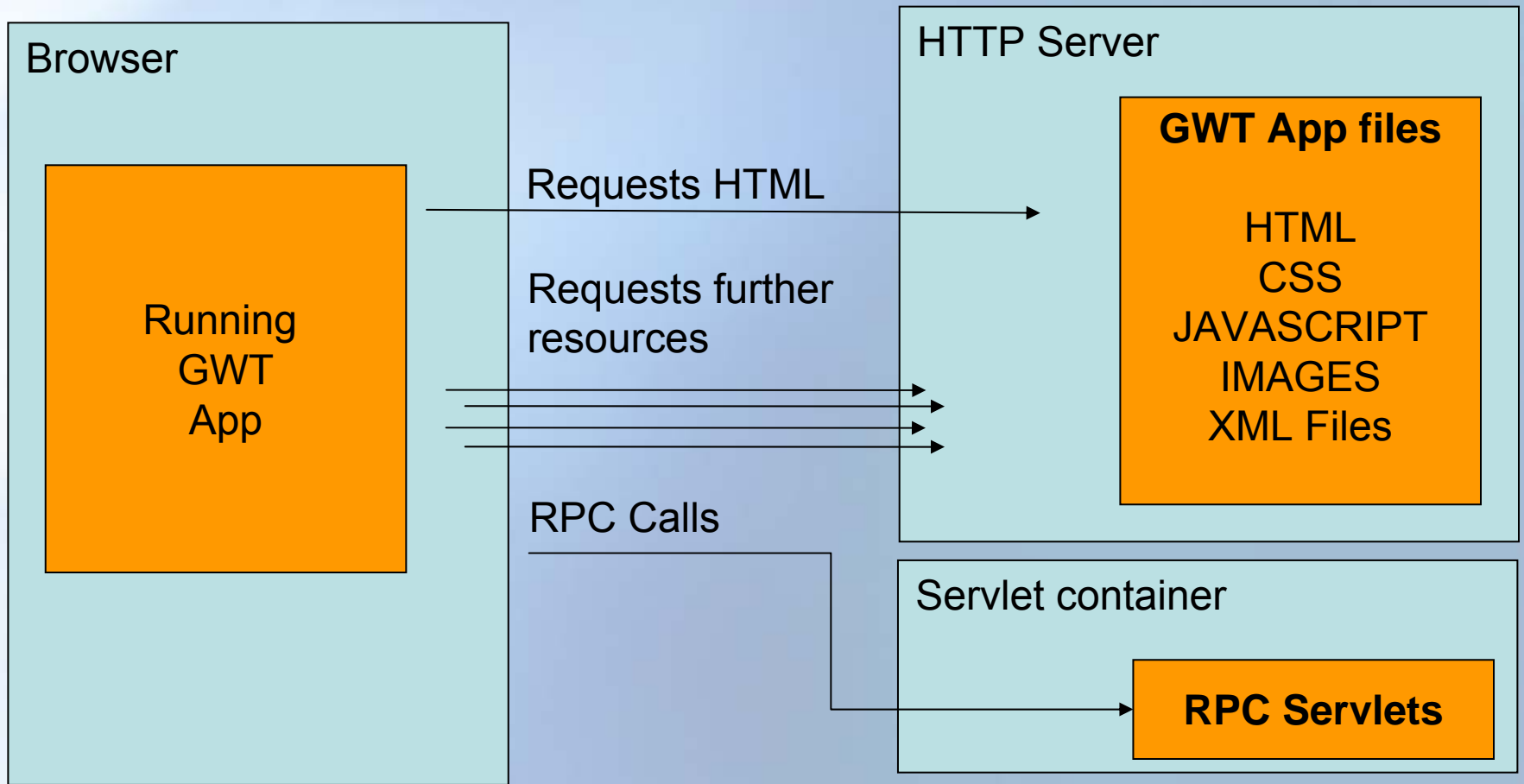
# GWT development cycle #3

- Confirm that your application works in **each browser** that you want to support, which usually takes no additional work.

# Developing GWT apps



# GWT runtime environment...



# Mission statement

- GWT's mission is to radically improve the web experience for users by enabling developers to use existing Java tools to build no-compromise AJAX for any modern browser.

# Demo

- Just a simple Hello World...
  - Show project structure
  - Show module descriptor
  - Show development shell
  - Show the application
  - Show the debugger

# Some more information

- It's not beta.
- It works.
- It is being used in production
  - Public sites
  - Many success stories
- Apache 2.0 License
- Actual version: 1.4

# GWT docs & co.

- Documentation could be better
- Many books available
- Many tutorials available
- Large community
- Many extensions

# Is it Java or not?

- Ok, what is Java?
  - Programming language (syntax)
  - Runtime environment (JVM)
  - Security concept
- GWT uses the Java programming language. Is it Java?

# Session topics

- Web 2.0, Ajax
- GWT – What is it?
- **Java EE and the Web**
- GWT and Java EE
- JSF done right?

# Java EE and the web

- Fact: web is first class citizen in the specification:
  - Servlets and JSPs spec
  - JSF spec
- Problem: it is not enough!
  - No application framework (web apps are apps!)
  - There no such thing as a simple web app

# Open source to the rescue

- Battle of frameworks
  - Been there, done that, came back and survived
- We still compare apples with oranges
  - Btw, what is a web framework?



# What do we have?

- Plain vanilla servlets & JSP
- Proprietary framework?

# Can we have a little more?

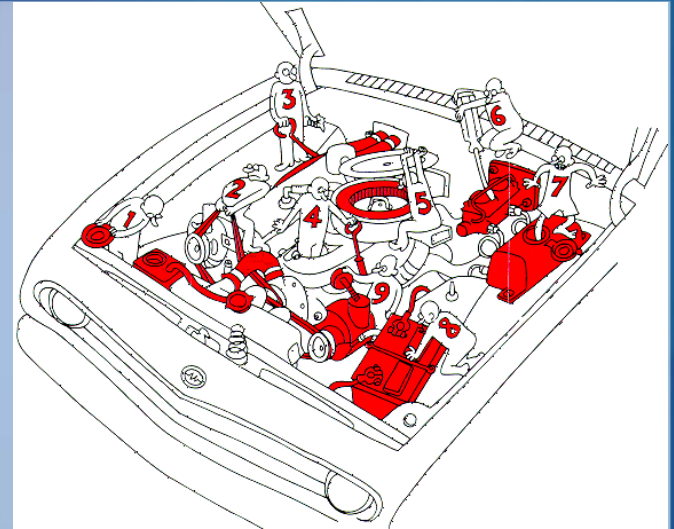
- MVC frameworks
  - Spring MVC
  - Struts (Struts 2)
- Template based frameworks
  - Wicket?
- Cool frameworks
  - Ruby on Rails?

# Can I have UI components?

- Component based frameworks
  - JSF
  - GWT
  - Echo2
  - Eclipse Rap

# All the same?

- Client side vs. server side rendering
- Single page vs. page roundtrip
- Little bit of Javascript vs. Javascript application
- Client side application vs. server side application (HOPE)



# Not really the same...

- Under the hood, each framework...
  - solves different technological issues
  - was needed by different people/ projects
  - started with a different historical background
  - started with a different set of technologies and ideas available

# Not the same...



# Again, what do we have?

- We have many different frameworks
- None of them has made it
- JSF is part of Java EE 5
  - JSF is a spec, not a product.  
Even JSF is not equal JSF
- Under the hood there are huge differences
- We have the choice, we must decide

# Session topics

- Web 2.0, Ajax
- GWT – What is it?
- Java EE and the Web
- **GWT and Java EE**
- JSF done right?

# EE developer point of view

- GWT development is Java development
  - Java coding (code completion, type safety)
  - Java tooling (the best IDE is the IDE you know best)
  - We can leverage our knowledge
    - Let's use patterns
    - Let's use refactoring
    - Let's do debugging
    - Let's do testing

# IDE – our day by day environment

- Don't change your environment
  - GWT uses Java
- Don't wait until someone writes some plugin for some new
  - Dynamic language
  - New XML description language

# GWT is extensible #1

- GWT knows something called modules
  - Young (huge, active) community developing many modules
    - Logging, drag and drop, UI components, etc...
- Write your own UI Components
  - By composition
  - By using JSNI

# GWT is extensible #2

- GWT let's you write your own generator
- GWT 1.5 provides new linker design, where you can change the compiler output

# GWT is more than a compiler

- GWT provides performance twirks that would be hard to do by hand
  - Image bundles
  - Ressource bundles (sandbox)
  - RPC optimization
  - Code optimization
    - Remove unused code
    - Code compression/ obfuscation

# Standalone application

- GWT application are standalone rich client applications running in the browser
  - Can use RPC to fetch/ send data to the server, as rich clients do.

# Server side GWT

- GWT provides a simple and powerful RPC framework
  - Server side is backed by servlets
  - Code gets optimized by the compiler
  - Sends serializable Java objects over the wire (between browser and server)
- Simple integration to Java backends

# Building GWT apps

- The GWT compiler is written in Java...
  - Command line tools
  - ANT scripts
  - MAVEN plugins
  - IDE plugins

# Session topics

- Web 2.0, Ajax
- GWT – What is it?
- Java EE and the Web
- GWT and Java EE
- **JSF done right?**

# What do we like in JSF?

- UI components, cool
  - Listeners
  - No more request-response actions
- Nice bean-bindings
  - Use EL in JSP/ XML to bind bean properties to fields
  - Validation framework
  - Converters

# Facelets – UI composition for non believers

- Facelets removes JSP from the technology stack
  - Thus removing lots of lifecycle problems
- Introduces easy composite pattern
  - Reuse of UI components
  - New components by aggregation

# Webflow – navigation for non believers

- Webflow provides a better navigation engine for JSF
- Provides new lifecycle/ scopes
- Solves back button problem
- Many more...

# What is not so nice...

- UI components jungle
  - Different programming models
  - Compatibility issues
- JSP/ Facelets XML
  - Need better tooling
  - Efficiency?

# What is not so nice...

- Server side rendering...
  - Latency: unhappy users
  - Resource consumption: unhappy service provider
- Know-How
  - Hard to learn?
  - Too much to learn?
- Missing application framework?

# And GWT?

- UI Components, really cool
- UI composition with Java is no issue
- Navigation is not an issue (single page principle)
- No server side rendering (happy user, happy service provider)
- Easy to learn (really!)

# GWT is missing

- A good databinding solution
- A good application framework for “simple applications”

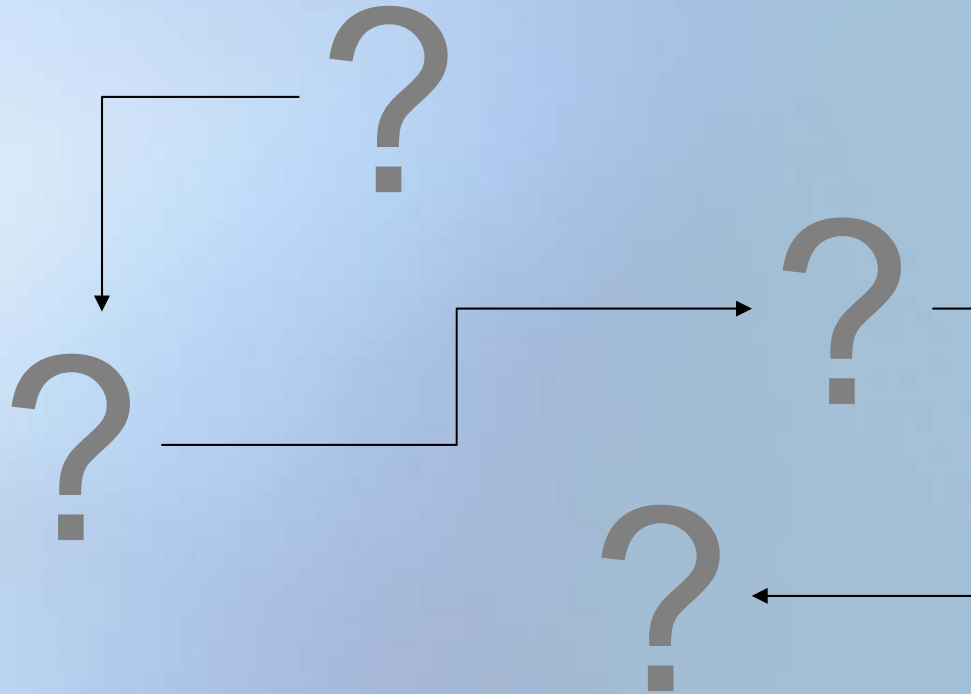
# Session topics

- Web 2.0, Ajax
- GWT – What is it?
- Java EE and the Web
- GWT and Java EE
- JSF done right?

# Session topics review

- Web 2.0, Ajax
- GWT – What is it?
- Java EE and the Web
- GWT and Java EE
- JSF done right?

# Time for questions?



# Thank you!

- Papick G. Taboada
  - Technology Scout
  - Software Architect

- Presentation material and more here:  
<http://adminsight.de>

