

PHP vs. Java

*This document reflects my opinion about PHP and Java. I have written this without any references. Let me know if there is a technical error.
--Hasari Tosun

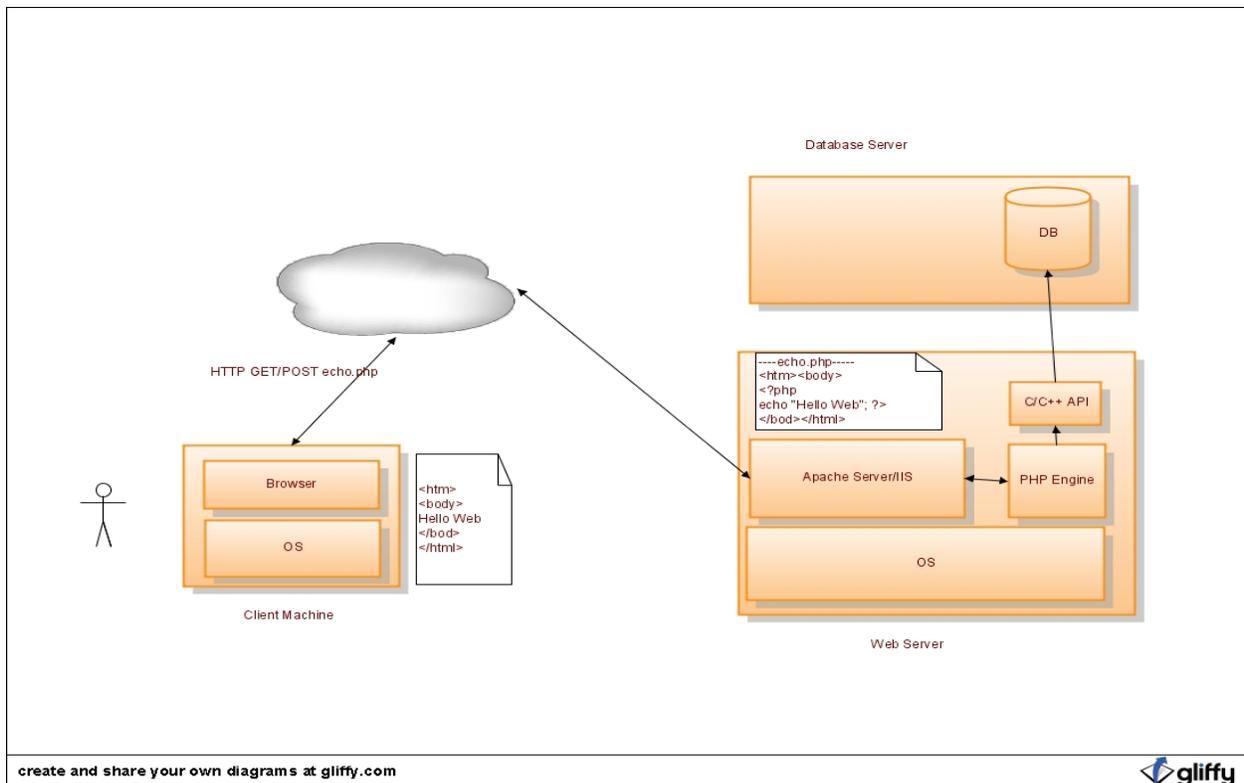
It isn't correct to compare Java to PHP. Since PHP is a **server-side** scripting language whereas Java is a general-purpose language. In other words, PHP is only used as a server-side language where Java is both for server-side and **desktop programming** language. Moreover, Java is **compiled** and **strongly-typed** language. On other hand, PHP is a **dynamic typed** language. Hence, only for server-side programming, the comparison between Java and PHP makes sense.

In this paper, I am not discussing following two issues since each is currently hotly debated in various communities:

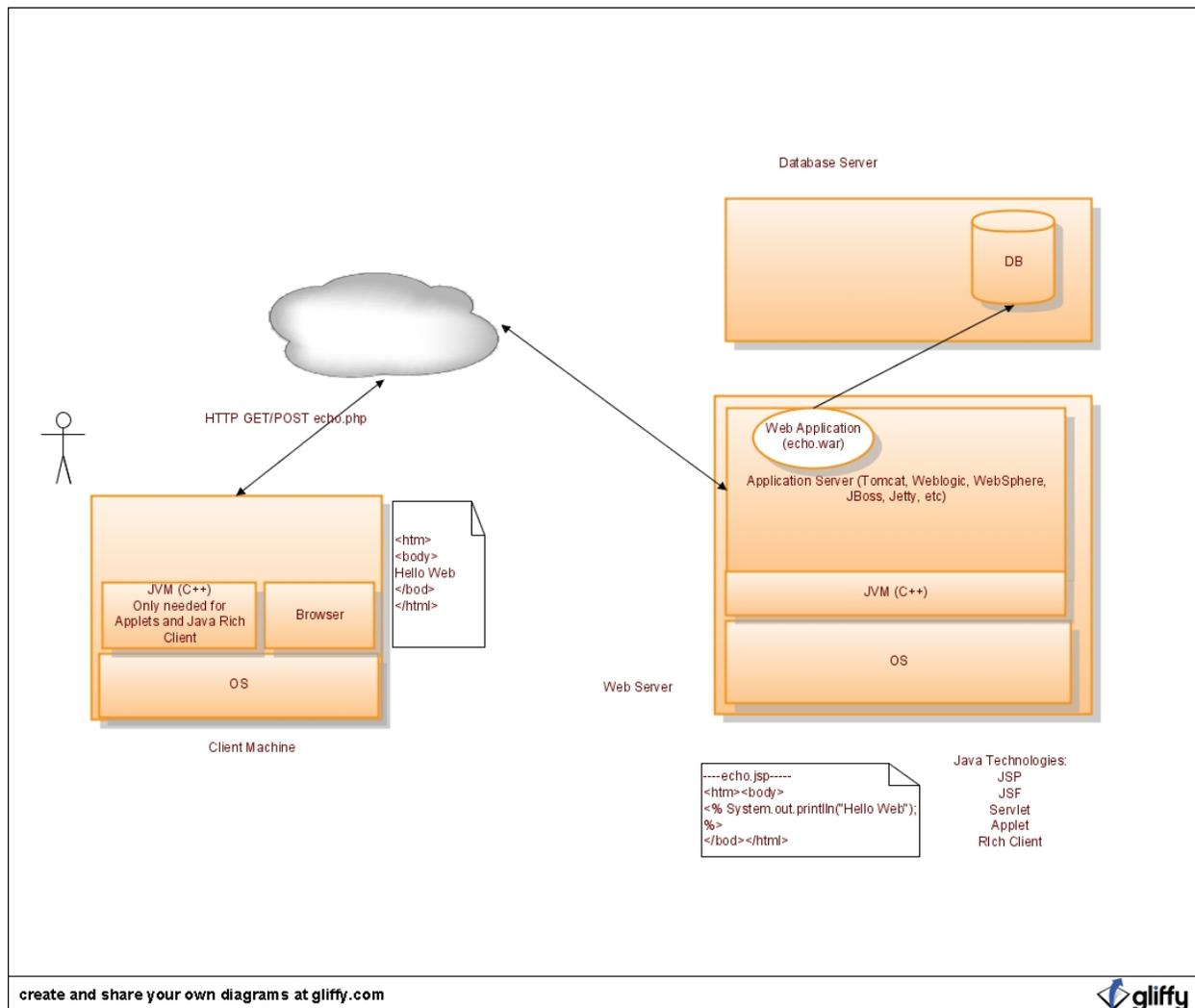
- **Strongly typed languages vs. dynamic Languages or Scripting**
- **Ajax vs. Smart Clients**

Web Architecture

Web Architecture for PHP Programmer



Web Architecture for Java Programmer



Programmers

- ❖ There are millions of bad programmers: Both Java and PHP programmers.
- ❖ Everybody is a PHP programmer. Even monkeys! Almost all web hosting companies use Apache Server/PHP.
- ❖ Changing or creating a page in PHP is easy:
Updating a page: `<?php echo "Hello Web" ?>` to `<?php echo "Hello Web 2.0" ?>`
Creating a page: Use *Drupal*, *WordPress*, *Xoops*, *Php-Nuke* etc
- ❖ For JSP (Java) it is also easy to change or create a page. However, for a complex application, monkeys cannot do programming in Java.

Language features & libraries

- ❖ Java is **OO**. So is PHP now (PHP 5.0).
- ❖ Java Provides **reflection** (reverse-engineer classes, interfaces, functions). PHP is finally added the same feature in PHP 5.0. **Note:** Reflection enable you to architecture your application better.
- ❖ Java has a richer set of **API**.
- ❖ Java provides management API (**JMX**) for managing and monitoring devices and applications.
- ❖ There thousands of **OSS** projects for Java (*apache.org*). Java provides a clean mechanism to combine these libraries (jar) to compose a complex application.
- ❖ Java provides native **transaction mechanism**.
- ❖ Java has a **persistence mechanism** to convert Java objects to database table entries and table entries to Java objects.
- ❖ Java has **annotations**: Annotations can be used for both compile-time and run-time.
- ❖ Java has **thread** support: Executing tasks in parallel.
- ❖ There are many other languages that can run within Java or on JVM (Ruby, Groovy, Python, **JavaFX** etc).

Language Tools

Editors

- Java has many fabulous editors: Eclipse, Netbeans and IntelliJ. For the productivity and the management of big project code, a complex editor is essential.
- PHP has a plug-in for Netbeans and Eclipse but very limited.

Debugging

- All Java editors allow you to debug your project within the editor.
- PHP debugging is just "**print**" statements.

Packaging and Deployment

- Java has many packaging and deployment utilities: Ant, Maven, Editors, Web Start etc.
- Java has packaging requirements: Web archive Files (war), Java Archive Files (jar).
- Compilation/Packaging protects your intellectual properties (**IP**)
- PHP are just files. No packaging concept.

Security

- ❖ Java Application servers enforce security model. Each application runs in its own container.
- ❖ Java/JVM has its own extensive security framework.
- ❖ No security guarantees from PHP: Administrators run PHP in CGI mode in order to avoid security nightmare. That is, PHP engine is **forked** for each page hit.
- ❖ Even Administrator **chroots** (each time to you hit a page) for security.

Performance

- ❖ JVM is optimized for performance and memory footprint: Just- In-Time-Compilation (**JIT**).
- ❖ Java compiles JSP files and JITs **hot spots** automatically.
- ❖ PHP parses the file and output the stream for each call. No compilation. Furthermore, in CGI model compilation doesn't make sense.
- ❖ JVM has **garbage collection (Memory Management)**.
- ❖ JVM has **object caching** mechanism.
- ❖ In Java Application Server, you could do **connection-pooling**. That is, a pool of connections can be shared by different client requests.
- ❖ In Java, the **session** is maintained on the server for the client where in CGI model session is destroyed each time.

Web 2.0+ Language Requirements

Feature	Java	PHP
<i>Garbage Collection</i>	Native Support	None
<i>Object Caching</i>	Native Support	None
<i>Re-targetable (Mobile, Desktop, Web etc).</i>	JVM	None
<i>Massively Distributed</i>	There are a few Java OSS: JavaSpace, Hadoop. All application servers support clustering.	None
<i>Persistent Session</i>	Native Support	<i>mod_php</i> in Apache. But, no one tries it for security
<i>JIT</i>	JVM support	None
<i>Connection Pooling</i>	Native Support	None
<i>Rich API</i>	Too much	Very little
<i>Security</i>	Native Support	
<i>Packaging and deployment</i>	Native Support	None
<i>Debugging</i>	Native Support	None
<i>Editors</i>	Many	Primitive
<i>Tools</i>	Many	Very few
<i>Persistence Mechanism</i>	Native Support	Third party libraries
<i>Transaction Mechanism</i>	Native Support	None
<i>Reflection</i>	Native Support	Native Support
<i>Application Management API</i>	Native Support	None
<i>OO for a sound Architecture</i>	Native Support	Limited Native Support
<i>Rich UI support</i>	Applet, Swing, JSF, JavaFX but not quite right	None
<i>Annotations</i>	Native Support	None
<i>Internationalization(i18N)/ Localization(i10N)</i>	Native Support	Not sure
<i>Threads</i>	Native Support	None