

# Transgression forswear-maintaining centralized database essay



**ASSIGN  
BUSTER**

Initially the language was called as “ oak” but it was renamed as “ Java” in 1995. The primary motivation of this language was the need for a platform-independent (I. E. , architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices. Java is a programmer’s language. Java is cohesive and consistent. Except for those constraints imposed by the Internet environment, Java gives the programmer, full control. Finally, Java is to Internet programming where C was to system programming. 2. 1. 1 Importance of Java to the internet: Java has had a profound effect on the Internet.

This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self- executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program Java can be used to create two types of programs: Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer.

It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java - compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and

dynamically change. 2. 1. 2 Features of Java Security Every time you that you download a “ normal” program, you are risking a viral infection.

Prior to Java, most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a “ firewall” between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or luscious intent. 1. Portability For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed . As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient. 2. The Byte code The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code.

Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-

time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on- the-fly compilation of byte code into native code.

Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JAVA, it compiles yet code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution. 3. Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology.

The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code. Java programming uses to produce byte codes and executes them.

The first box indicates that the Java source code is located in a .java file that is processed with a Java compiler called javac. The Java compiler produces a file called a .class file, which contains the byte code. The .class file is then

loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code. Fig 2. 1 Picture showing the development process of JAVA Program 4. Java Architecture Java architecture provides a portable, robust, high performing environment for development.

Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet. 5. Compilation of code When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines.

This machine is called Java Virtual Machine. Fig 2. 2 Compiling and interpreting Java Source Code During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Sun's station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets. 2. 2 SALIENT FEATURES OF JAVA 2. 2. 1 Simple: Java was designed to be easy for the Professional programmer to learn and to use effectively.

If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object

oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

2. 2. 2 Object-oriented: Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects.

The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high- performance non-objects.

2. 2. 3 Robust: The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time. Java virtually eliminates the problems of memory management and delineation, which is completely automatic.

In a well-written Java program, all run time errors can -and should -be managed by your program.

2. 3 HYPER TEXT MARKUP LANGUAGE Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks). HTML is not a programming language but it is an application of ISO Standard 8879, SUMS (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily Jump from one point to another point.

We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized words that load to there documents or some portions of the same document. HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc. , can enhance the presentation of the document. Anything that is not a tag is part of the document itself. 2. 3. 1 Advantages of HTML: A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information. HTML is platform independent. HTML tags are not case-sensitive. Easy to learn & code even for novice programmers Supported on almost every browser, if not all browsers.

Widely used; established on almost every website, if not all websites. 2. 3. 2 Basic HTML tags: Creates hypertext links. Formats text as bold. Contains all tags and text in the HTML document. . Definition off term. Creates definition list. Formats text with a particular font. Encloses a fill-out form. Defines a particular frame in a set of frames. . Creates headings of different levels. Aligns text to center. Contains tags that specify information about a document. Other HTML tags. Provides meta-information about a document. Contains client-side or server-side script.

Contains all Creates a table. . Indicates table data in a table. Designates a table row. 2. 4 JAVASCRIPT JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page.

On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display accordingly Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags JavaScript statements 2. 4. 1 A few things that we can do with Java script: contents of a form and make calculations.

Validate the Add scrolling or changing messages to the Browser's status line. Animate images or rotate images that change when we move the mouse over them. Detect the browser in use and display different content for different browsers. Detect installed plug-ins and notify the user if a plug-in is required. We can do much more with JavaScript, including creating entire application. 2. 4. 2 JavaScript vs.. Java: JavaScript and Java are entirely different languages. A few of the most glaring preferences are: Java applets



are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.

While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications. There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

2. 4. 3 Advantages: JavaScript can be used for Sever-side and Client-side scripting. It is more flexible than ascent. JavaScript is the default scripting languages at Client-side since all the browsers supports it. . 5 JDBC

Java Database Connectivity or in short JDBC is a technology that enables the Java program to manipulate data stored into the database. Here is the complete tutorial on JDBC technology.

2. 5. 1 What JDBC? JDBC is Java application programming interface that allows the Java programmers to access database management system from Java code. It was developed by Java Soft, a subsidiary of Sun Microsystems. JDBC has four Components: 1. The JDBC API. 2. The JDBC Driver Manager. 3. The JDBC -rest suite. 4. The JDBC-ODBC Bridge.

The JDBC API The JDBC application programming interface provides the facility for accessing the relational database from the Java programming language. The API technology provides the industrial standard for independently connecting Java programming language and a wide range of databases. The user not only execute the SQL statements, retrieve results, and update the data but can also access it anywhere within a network because of it's " Write Once, Run Anywhere" (WORD) capabilities.

Due to JDBC API technology, user can also access other tabular data sources like spreadsheets or flat files even in the a heterogeneous environment.

JDBC application programming interface is a part of the Java platform that has included Java Standard Edition (Java SE) and the Java Enterprise Edition (Java EE) in itself. The JDBC API has four main interfaces: The latest version of JDBC 4.0 application programming interface is divided into two packages. 1. `java.sql`: `java.sql`. Java SE and Java EE platforms are included in both the packages. 2. The JDBC Driver Manager: The JDBC Driver Manager is a very important class that defines objects which connect Java applications to a JDBC driver.

Usually Driver Manager is the backbone of the JDBC architecture. It's very simple and small that is used to provide a means of managing the different types of JDBC database driver running on an application. The main responsibility of JDBC database driver is to load all the drivers found in the system properly as well as to select the most appropriate driver from opening a connection to a database. The Driver Manager also helps to select the most appropriate driver from the previously loaded drivers when a new open database is connected. 3.

The JDBC Test Suite: The function of JDBC driver test suite is to make ensure that the JDBC drivers will run user's program or not. The test suite of JDBC application program interface is very useful for testing a driver based on JDBC technology during testing period. It ensures the requirement of Java Platform Enterprise Edition (JEE). 4. The JDBC- ODBC Bridge: The JDBC-ODBC bridge, also known as JDBC type 1 driver is a database driver that utilize the

ODBC driver to connect the database. This driver translates JDBC method calls into ODBC function calls.

The Bridge implements Jdbc for any database for which an Odbc driver is available. The Bridge is always implemented as the sun. Jdb. ODBC Java package and it contains a native library used to access ODBC. 2. 5. JDBC Architecture 1. Database connections 2. SQL statements 3. 4. 5. 6. 7. 8. 9. Result Set Database metadata Prepared statements Binary Large Objects (BLOBS) Character Large Objects (Slobs) Callable statements Database drivers 10. Driver manager The JDBC API uses a Driver Manager and database-specific drivers to provide transparent connectivity to heterogeneous databases.

The JDBC driver manager ensures that the correct driver is used to access each data source. The Driver Manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases. The location of the driver manager with respect to the JDBC drivers and the service is shown in Figure: Layers of the JDBC Architecture Fig 2. 3 JDBC Architecture A JDBC driver translates standard calls JDBC into a network or database protocol or into a database library API call that facilitates communication with the database.

This translation layer provides JDBC applications with database independence. If the back-end database changes, only the JDBC driver need be replaced with few code modifications required. There are four distinct types of JDBC drivers. 2. 5. 3 JDBC Driver and Its Types Type 1 JDBC-ODBC Bridge: Type 1 drivers act as a “ driver” between JDBC and another database

connectivity mechanism such as ODBC. The JDBC-ODBC bridge provides JDBC access using most standard ODBC drivers. This driver is included in the Java 2 SD within the sun. jdbc. ODBC package.

In this driver the Java statements are converted to jdbc statements. A JDBC statement calls the ODBC by using the JDBC- ODBC Bridge. And finally the query is executed by the database. This driver has serious limitation for many applications

Type 1 JDBC Architecture Fig 2. 4 Type 1 JDBC Architecture

Type 2 Java to Native API: Type 2 drivers use the Java Native Interface (JNI) to make calls to local database library API. This driver converts the JDBC calls into a database specific call for databases such as SQL, ORACLE etc. This driver communicates directly with the database server.

It requires some native code to connect to the database. Type 2 drivers are usually faster than Type 1 drivers. Like Type 1 drivers, Type 2 drivers require native database client libraries to be installed and configured on the client machine.

Type 2 JDBC Architecture Fig 2. 5 Type 2 JDBC Architecture

Type 3 Java to Network Protocol or All- Java Driver: Type 3 drivers are pure Java drivers that use a proprietary network protocol to communicate with JDBC middleware on the server. The middleware then translates the network protocol to database-specific function calls.

Type 3 drivers are the most flexible JDBC solution because they do not require native database libraries on the client and can connect to many different databases on the back end. Type 3 drivers can be deployed over the Internet without client installation. Java → JDBC statements → SQL

statements databases. Type 3 JDBC Architecture Fig 2. 6 Type 3 JDBC

Architecture Type 4 Java to Database Protocol: Type 4 drivers are pure Java drivers that implement a proprietary database protocol (like Oracle's SQL\*Net) to communicate directly with the database.

Like Type 3 drivers, they do not require native database libraries and can be deployed over the Internet without client installation. One drawback to Type 4 drivers is that they are database specific. Unlike Type 3 drivers, if your back-end database changes, you may have to purchase and deploy a new Type 4 driver (some Type 4 drivers are available free of charge from the database manufacturer). However, because Type 4 drivers communicate directly with the database engine rather than through middleware or a native library, they are usually the fastest JDBC drivers available.

This driver directly converts the Java statements to SQL statements. Type 4 JDBC Architecture Fig 2. 7 Type 4 JDBC Architecture So, you may be asking yourself, " Which is the right type of driver for your application? " Well, that depends on the requirements of your particular project. If you do not have the opportunity or inclination to install and configure software on each client. However, if the cost of Type 3 or Type 4 drivers is prohibitive, Type 1 and type 2 drivers may become more attractive because they are usually available free of charge.

Price aside, the debate will often boil down to whether to use Type 3 or Type 4 driver for a particular application. In this case, you may need to weigh the benefits of flexibility and interoperability against performance. Type 3 drivers offer your application the ability to transparently access different types of

databases, while Type 4 drivers usually exhibit better performance and, like Type 1 and Type 2 drivers, may be available free of charge from the database manufacturer. 2. 6 JAVA SERVER PAGES (JSP) Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages.

Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches; it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work. 2. 6. Features of JSP Portability: Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifestyle and its interaction components. Components It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file.

The components currently supported include Java Beans, and Servlets.

Processing A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet <https://assignbuster.com/transgression-forswear-maintaining-centralized-database-essay/>

that is generated outputs real content in straight HTML for responding to the client. Access Models: A Java Server Pages file may be accessed in at least two different ways.

A client's request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client. In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

Steps in the execution of a JSP Application: 1) The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page. 2) This request is transferred to the Servlet container. At the server side Servlet container receives the request and if it is a request for a JSP file server gives this request to the JSP engine. 3) JSP engine is a program which can understand the tags of the JSP and then it converts those tags into a Servlet program and it is stored at the server side.

This Servlet is loaded in the memory and then it is executed and the result is given back to the Servlet container and then it is transferred back to the client. 2. 6. 2 Java Beans: We can assemble a computer or fan very easily by choosing different components manufactured by different vendors. We can take a

screw from company one and use it to fit the Mother board to cabinet as they are manufactured according to a tankard.

Observing to this point to simplify the process of developing software, different software companies has proposed different component technologies. Ex: java soft Java bean component tech, EJB component tech, Microsoft COM, BONO component model. Java Bean and EJB are two different specifications from Java soft. EJB can be used to implement business logic on the server side. Most of the developers uses to assume Java Bean components are for developing GUI components and they can be used only on the client side but we can develop any kind of software using Java Bean standard (GU'/ non GU').

Java Bean can be used either on the client side or on the server side. AWT, JEFF components are implemented according to Java Bean standard.

According to Java Bean standard a Bean component can support a set of properties, set of events, any number of additional methods. A property can be read-write or it can be just read only property. For read write property we need to provide sextet and getter methods (six if the property is Boolean ).

To support the following properties (I) name (it) email (iii) age according to Java bean standard we need to write the code as, public class Serbian  
{String name; String email; into age; public void sequestrate( String value )  
{name = value; } public void estimate( String value ) { email = value; }  
public void stage( into value ) { age = value; } public String stammering()  
{ return name; } public String Guatemala() { return email; } public into  
stage() { return age; }} Java Beans like Cobnuts supports the events by



providing the methods with naming patterns. I) dexterousness (it) remonstratively Apart from developing Java bean class we can also provide Beanpole class. In this class we can provide (I) Information about properties (it) Information (iii) Information about the icon that represents our bean. Iv) According to JSP model we can develop the application as, Fig. 8. Picture showing JSP model 1. According to above model the presentation logic has to be implemented in JSP page and the business logic has to be implemented as part of Java bean This model help us in separating the presentation and business logic.

For a large scale projects instead of using model it is better to use modems (MFC). Struts frame work is based on model 2. 2. 7 STRUTS What is Struts - Struts Architecture Struts is famous for its robust Architecture and it is being used for developing small and big software projects. Struts is an open source framework used for developing JEEZ web applications using Model View Controller (MVC) design pattern. It uses and extends the Java Servile API to encourage developers to adopt MVC architecture.