main index menu

XDisect
Super high speed XML indexing and searching

Version

Architectural Concept Overview

XDisect XML / HTTP Interface

Email eval@pybiz.com with questions or comments

The XDisect Application Server provides the services of an Internet repository using XML / SQL and HTTP. The combination of a flexible, evolving schema coupled with XDisect's flexible architectural approach provides unparalleled suport for challenging business conditions.

This document discusses some of the different application architectures that are feasible using the XDisect Application Server with XML and HTTP interfaces.

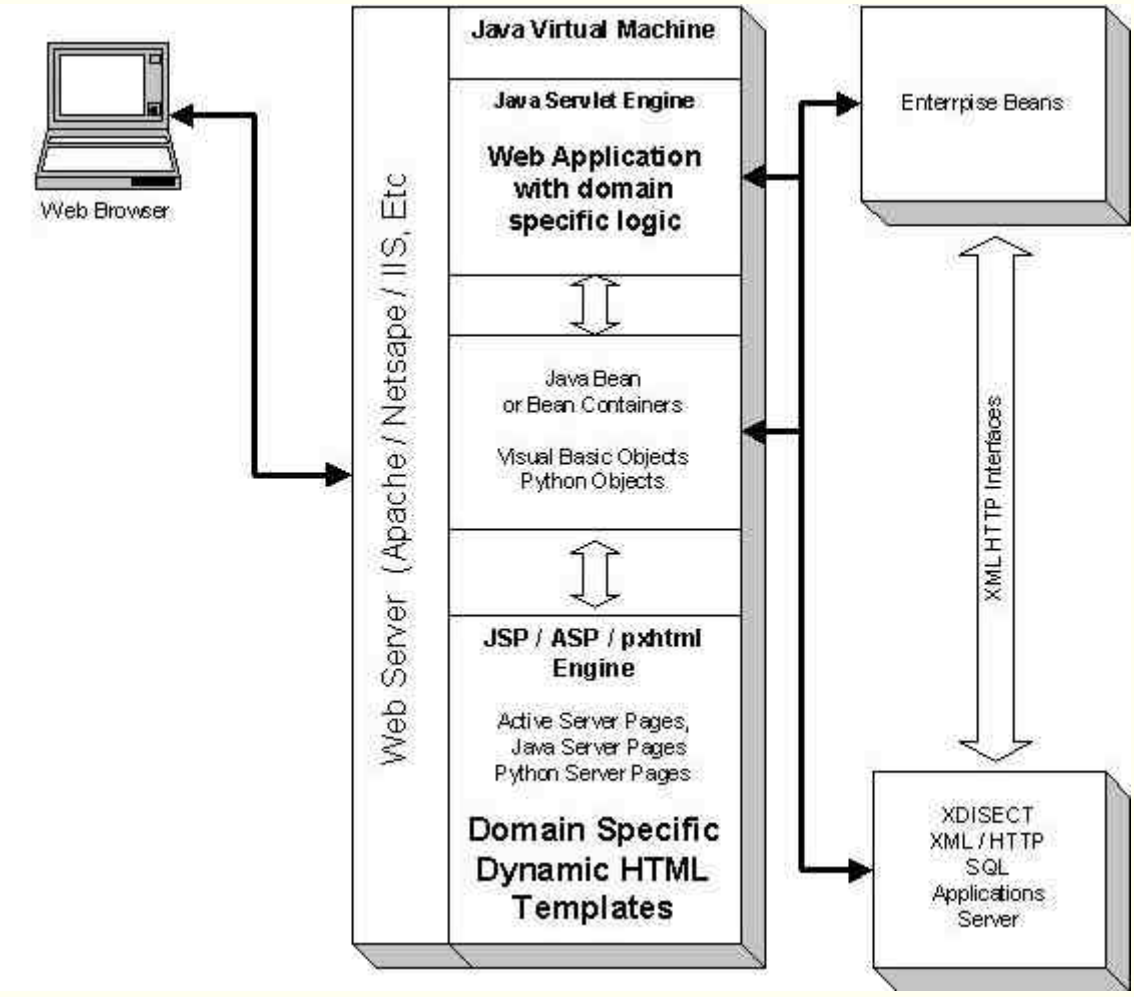
Note: It is becoming popular to mix and match architectural models in the same application. For example, using XSQL along with a servlet based application is a natural extension and ends up creating a hybrid that is stronger than either approach when used individually.

Servlet Based Web Application Using XDisect

This architectural model is a PyBiz favorite because it keeps layers isolated in a way that makes it feasible to pull components out and replace them with new components as technology matures.

This servlet architecture is interesting because many new applications are being developed using WebLogic, JSP and Enterprise Java Beans.

The XDisect Internet repository adds advanced features to this architecture that are capable of dealing with evolving and variable shemas. This extends the capability of the architecture to deal with changing business conditions while retaining its overall strength and encapsulation.



The requests from the browser arrive at either

- A registered servlet.
- Application Server Pages (JSP) which in turn use Beans to obtain sophisticated processing.

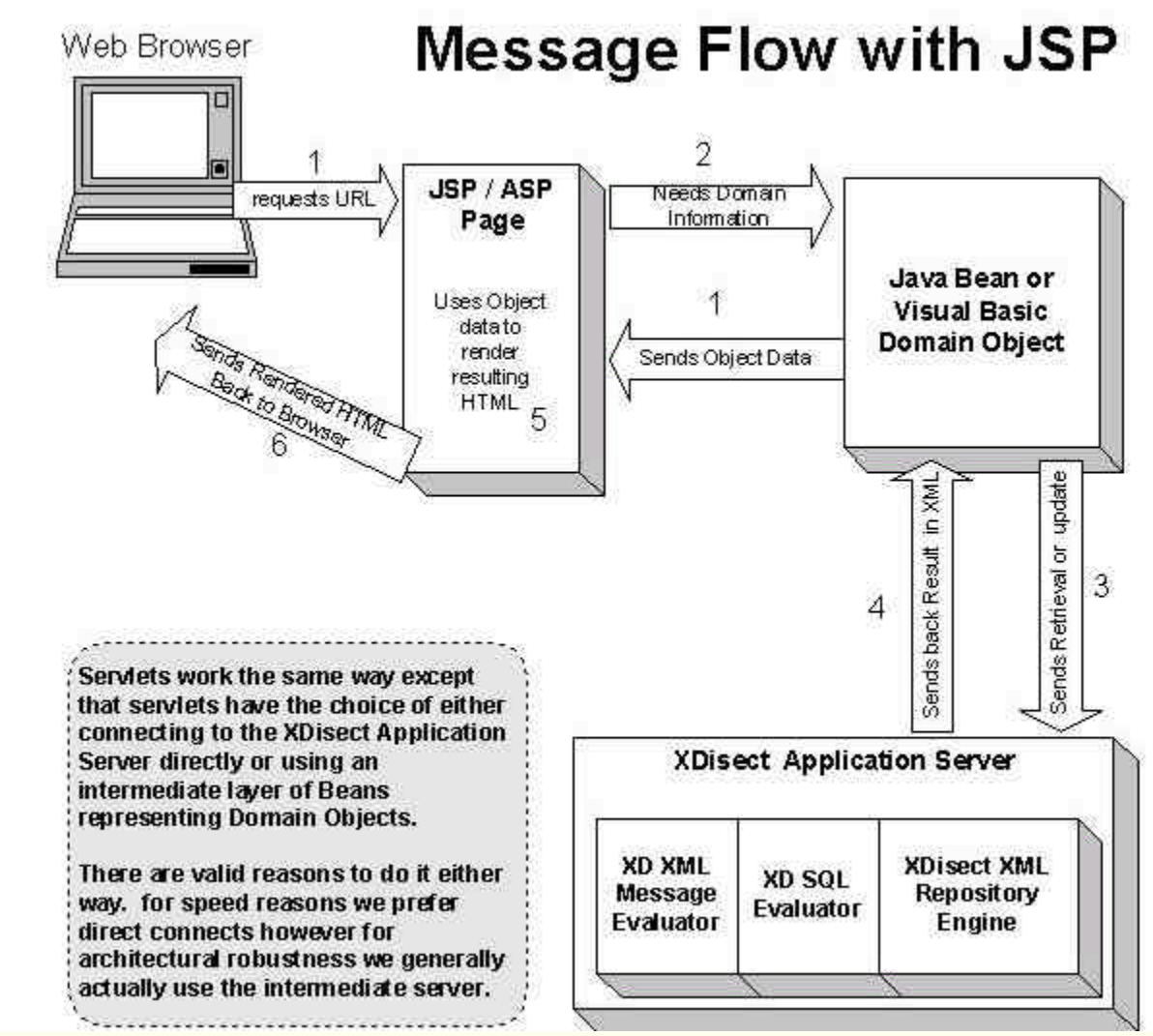
How requests are sent to the XDisect engine

- The Servlets can directly send XML HTTP requests to the XDisect Server.
- The Beans or (domain objects) can send requests to the XDisect Server.
- The Domain Templates (In some languages) talk to the servlet which then talks to the XDisect Server.

Any architectural layer for the specific application that needs to can access the XDisect repository using the XDisect Application Server.

Message Flow with JSP

Another way to look at the ASP / JSP web application is by looking at the sequence of actions while working on a request for a user.



- The interface between the bean object and the XML Internet repository (items 3 & 4) may actually make many round trips in the process of rendering a single page. One of the benefits of the design of the

XDisect XML / HTTP interface is that it can dramatically reduce network round trips.

- The XML / HTTP interface for XDisect is language neutral which means that domain objects built in any language can interact with it. This means that it is feasible for the JSP page to talk directly to the XDisect Application Server. The XDisect Application Server provides it's own HTTP server and Python servlet engine which run's in-process. Additional documentation on the XDisect Application Server and .pxhtml pages is available upon request.

XSQL & XSLT Transformation

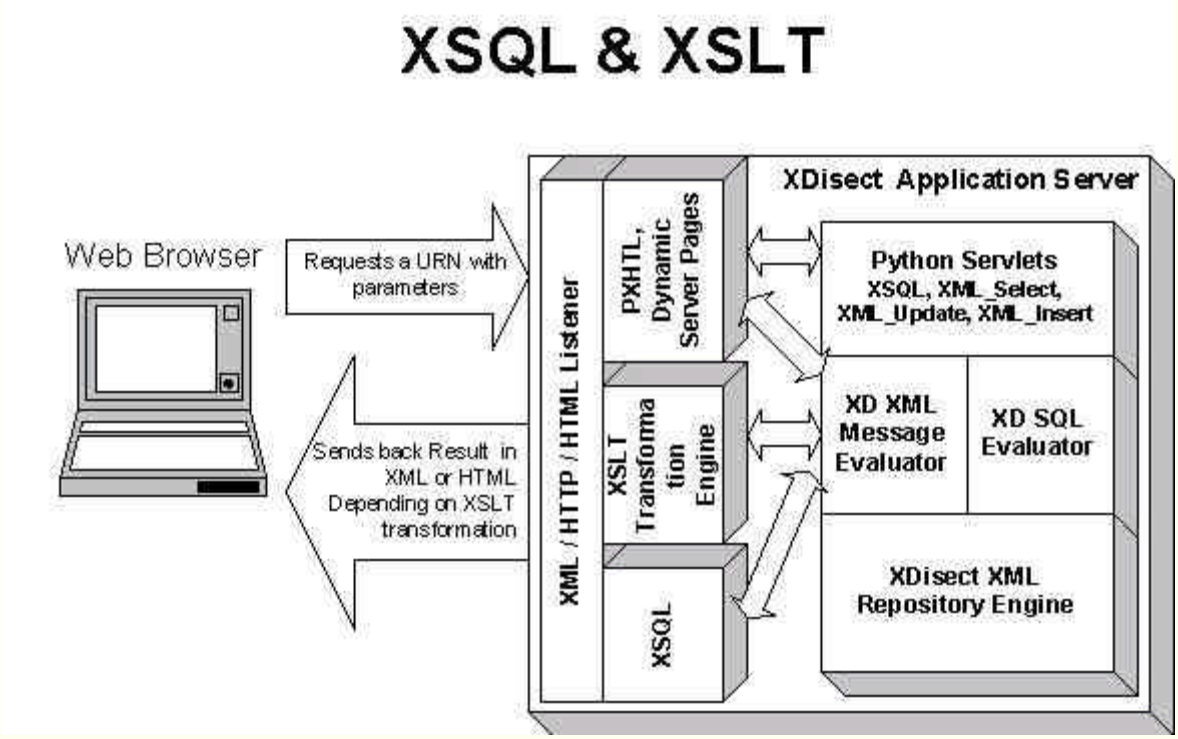
XSQL is an XML based SQL interaction standard that is heavily integrated with XSLT. Oracle introduced XSQL as a way to provide some XML capabilities in their RDBMS engine. XDisect supports XSQL..

With XSQL the actual SQL command is stored on the server. When a command arrives, the server loads that file and executes it directly.

- Server based templates make the XSQL interface somewhat more safe than full client/server implementations where the client application has the oportunity to create whatever SQL statement it wants. However, it is also less flexible for the same reason.
- The HTTP query parameters from the URL are extracted and interpolated into the XSQL command prior to submission to the XDisect Message Evaluator.
- XSQL is cool because it's core design makes it easy to supply basic parameters to the SQL statements and the client. Because the data parameters are passed as part of a URL string the application code never has to actually know what the original SQL looked like. This allows the SQL to be optimized at a later time without impacting the client.

◦ Some features were changed slightly in order to fully utilize the advanced capabilities of XDisect. Due to it's native XML strategy, XDisect has capabilities that are not feasible for normal relational engines. The syntax was modified where necessary to support these advanced capabilities.

XSLT transformation can be performed either on the server or on the client. Right now only Microsoft's IE5 does a credible job of XSLT in the browser.



- The Web browser could just as easily be another Web application.
- The selection of whether HTML or XML is sent back depends on the navigation structure.

Sample XSQL Statement

```
<?xml version="1.0"?>
<xsql:query connection="demo" xmlns:xsql="urn:pybiz-xsql">
   SELECT
   WHERE person*skill* CONTAINS {@skill}
     AND person*phone* CONTAINS {@area}
</xsql:query>
```

- We use the XSQL style {@skill} interpolation which really means to pull the skill variable from the URL query parm http://127.0.0.0:8060/select.2.xsql?skill=java?area=408.
- Note the use of person*skill in the fields "person.skill" and "person.job.skill" and "person.job.skill.programming.lang".
- XSQL provides an ideal interface for allowing external companies to make safe queries against existing repositories. Its server based templates make it quite easly to limit queries against the database to those wich have been tested as safe.

Direct Applet Access to the XDisect Engine

In the Direct Applet scenario almost all processing is done by the local client. This is interesting because it saves server resources while often

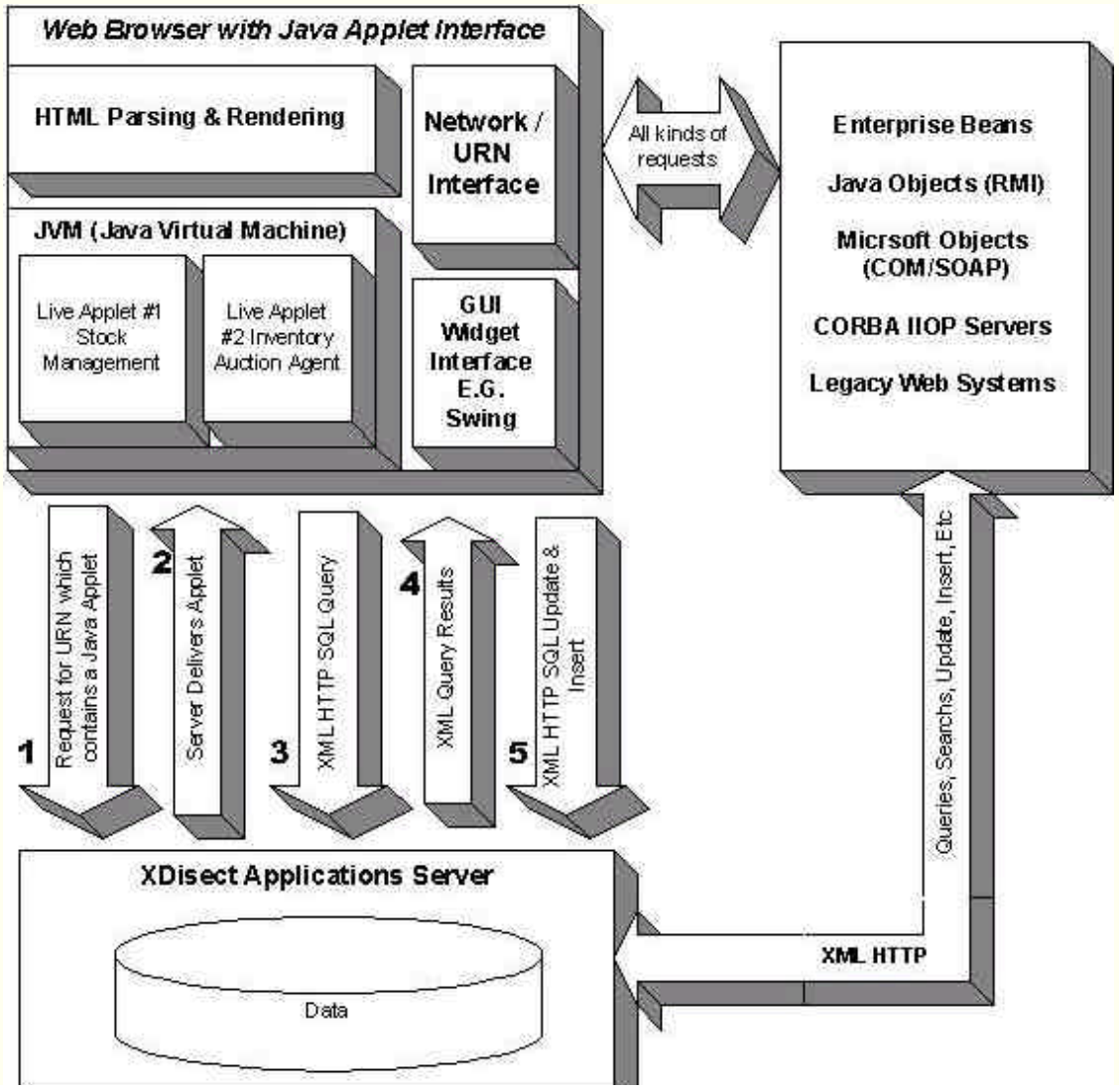
creating a more dynamic user experience.

Local browser GUI tool kits such as SWING have allowed engineers to overcome the visual design and user interaction limitations of the browser while preserving the software distribution advantages of the web. This is bound to become a very popular strategy and is likely to be combined with other architectural models.

Applet Platforms

- Java is by far the most popular applet environment.
- TCL/TK is also available as an applet environment.
- Smalltalk (Squeak) is available as an applet environment.
- Python is available as an applet environment.

Any applet platform can interact with the XDisect Applications Server as long as it supports a full XML parser and has the ability to talk over TCP/IP. Java is particularly good at both of these requirements.....



Java applets can connect to remote URL's as long as they are on the same server from which they were served. They can make IIOP, RMI and EJB calls to access enterprise resources. This flexibility makes them ideal to interact directly with the XDisect engine using the XML / HTTP interface.

CGI Scripting Integration, (Perl, Python, TCL, etc.)

CGI Scripting is the original approach to building web applications and if an accurate measurement were to be taken it is still the most widely deployed.

When using CGI techiques the Web Server spawns a separate process for each request.

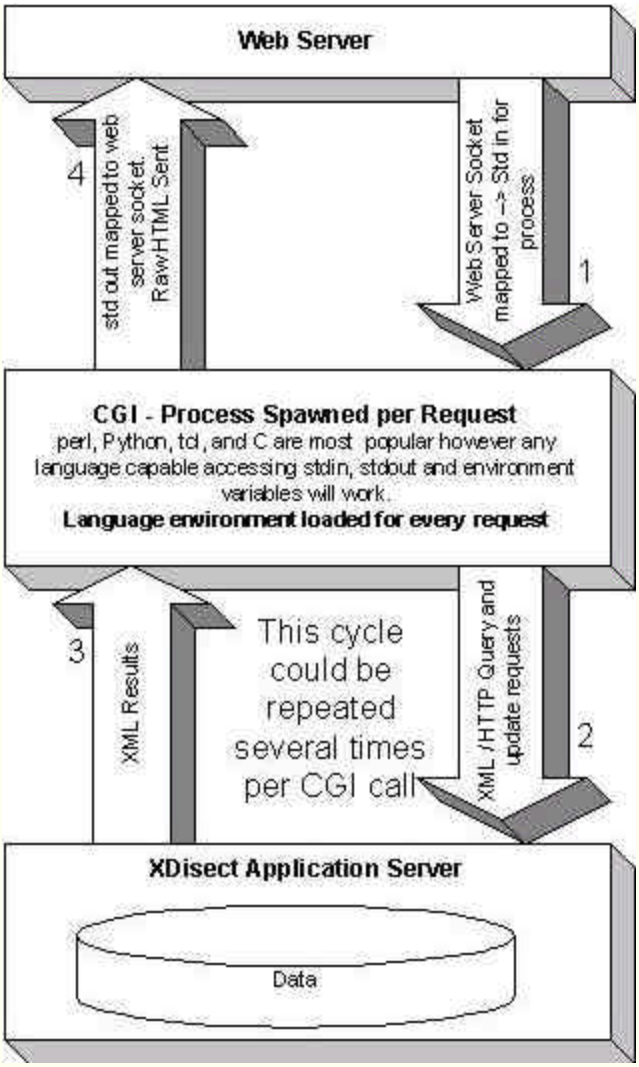
- Input and output to this sub-process is done by a combination of enviornment variables and remapping the stdin and stdout streams for the the process.
- The process spawn and load per request can get quite expensive and can end up limiting the ability of an application to scale. However there are many inovative ways to overcome this limitation.

CGI remains popular because of its resiliance.

- If the application script request crashes, it only takes down that one process and leaves the web server in a stable state.
- This can make it easier to ensure that most users get correct behavior even when there may be a bug or two in the software.

CGI Scripts must rebuild their state every time they are executed.

- XDisect makes this easier by providing easy access to complex domain ojbects using standard SQL embedded in XML.
- XDisect caches recent query results, so it is able to respond very rapidly with frequently requested objects even when they are complex.



See [XDisect Sample Clients] for examples of clients in most popular scripting languages.

PyBiz software is built and tested first on Linux so it runs best on Linux.

www.pybiz.com
eval@pybiz.com 408-364-1741

Powered By
PyBiz