

Application architecture and web interfaces

A common feature of many modern GUI applications is that there is a strong desire to deliver the applications via web browsers. MSIE and Navigator have differences in their implementation of (what should be) standard extensions such as Java, and so any web based developments need to be tested on any target platforms.

However - its not really that hard to write CGI/ PHP and/or Java applications that work on all platforms. In this section we will briefly look at CGI, PHP and Java.

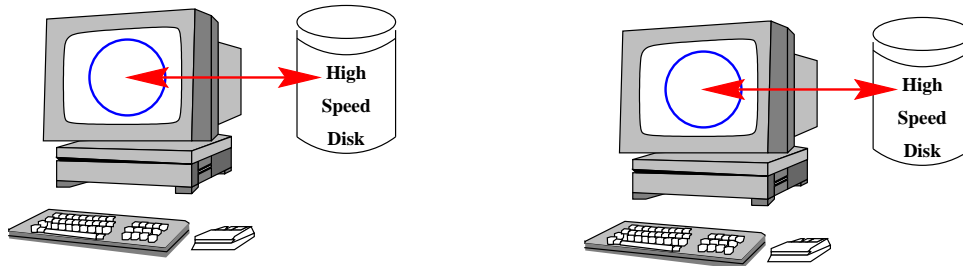
7.1 Architecture of GUI applications

We can categorise GUI applications in many different ways, but the overall communications architecture is of interest, and helps us select tools and development strategies. One classification is:

- Standalone
- Shared file
- Shared database
- Web based
 - Simple
 - Scripting
 - Java

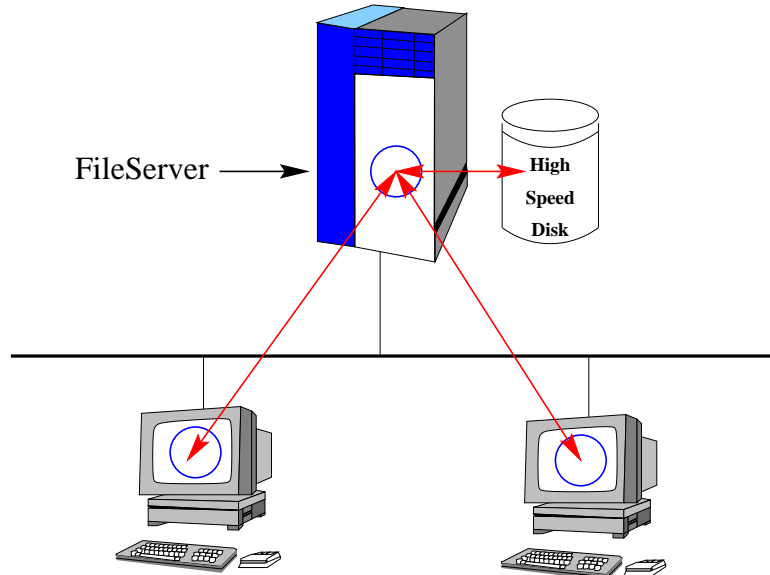
7.1.1 Standalone

A standalone GUI application runs on the user's PC, and reads and writes a local disk for files or (in a very general sense) databases. Note that Microsoft Access programs are normally of this sort.



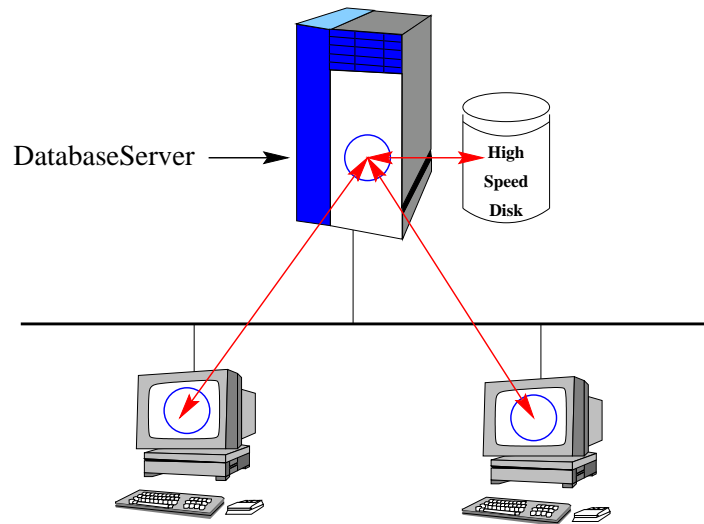
7.2 Shared file

A GUI application runs on the user's PC, and reads and writes a shared disk for files or (in a very general sense) databases. Note that Microsoft Access programs may be of this sort, but that Access files are generally not shareable.



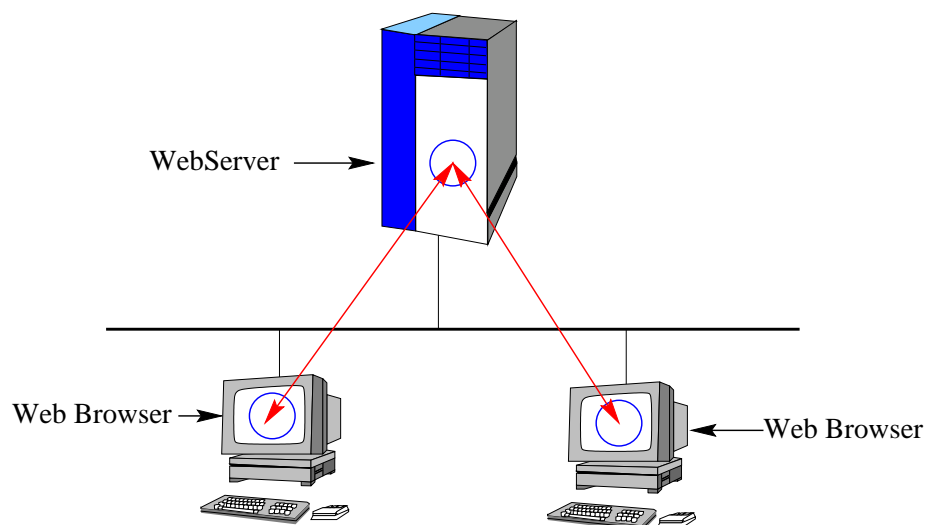
7.3 Shared database

A GUI application runs on the user's PC, and reads and writes a local database. Note that when your application uses a database like Oracle or Microsoft SQL server, it is likely to be of this sort.



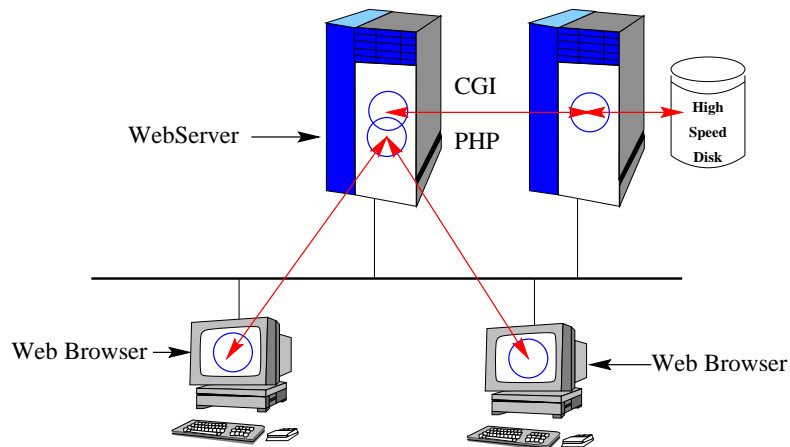
7.4 Web server applications

A VERY simple GUI application might be constructed using a series of interlinked web pages found on a server, and relying on a web browser on the client PCs. Help documentation applications are sometimes of this sort.



7.5 Web server with active scripting

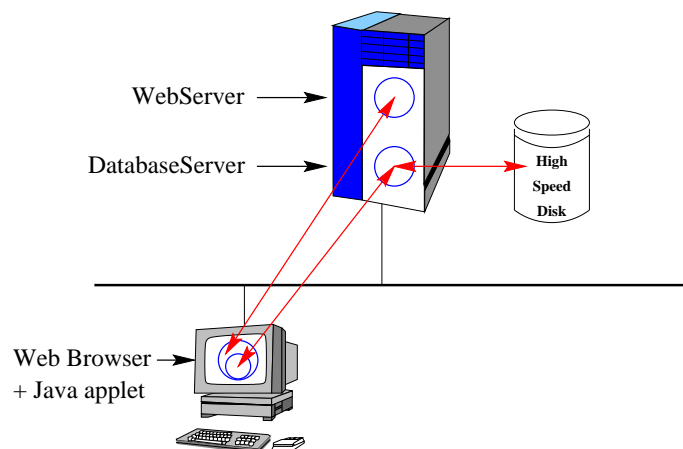
A more complex GUI application might be constructed using a series of interlinked web pages found on a server, and relying on a web browser on the client PCs. However, by using the Common Gateway Interface, or some other scheme of server-side scripting, we can return a program-derived web page to the application. CGI and PHP are both examples of this technique.



7.6 Web server with Java applet

An even more complex GUI application might be constructed using a series of interlinked web pages containing Java applets. The advantage of this, is two fold.

1. The processing load on the web server may be reduced.
2. The Java applet can directly¹ communicate with a database server.



¹Note that there are some security concerns here.

7.11 Assignment 4 - Implementation

In this assignment, you may work in a group of up to 4 people - or you may do it by yourself. The assignment is worth 40% of your assignment grade, and is due at 5:00 p.m. on Friday, 19th April, 2002 - please deliver to Hugh's room. The assignment is to be done using Java/Swing.

Task:

Your task is to implement and document a multi-user simple text editor. By this I mean that you and other people may safely simultaneously edit a single file (perhaps using a shared file system). Note that there is no one correct way of doing this, and feel free to experiment with different approaches.

You may re-use existing simple text editor code found in many introductory Java/Swing tutorials if you wish, but the extensions to make it multi-user must be your own.

Here are brief descriptions of possible approaches to this problem:

1. (Simplest) Each user edits their own copy of the file. When the user saves the file, other users are notified that the original file has changed, and given a choice to load the new changed file, or to just continue.
2. (Medium) Each user edits a part of the file, chosen when they open the file from the remaining editable parts (For example, lines 1-100 for user 1, lines 101-200 for user 2). When the user saves the (part) of the file, other users are notified that the original file has changed, and their programs automatically load the new changed parts.
3. (Very tricky) Each user edits their own copy of the file. As changes are made, all users screens are updated.

Feel free to dream up more advanced shared editing schemes. Note that since communication between the programs may be tricky, I suggest your programs communicate through another shared file.

Deliverables:

- A title page containing your names and matriculation numbers.
- A ten to twenty page document containing
 - A brief summary and justification of the overall strategy used for shared editing - perhaps with a state diagram describing the states of each edit program, with an argument as to why your program is "safe".

- An overview of the interface design
 - A user manual for the interface
- A disk containing the code, with a (small) README file to explain how I am supposed to run your software.

Note that this assignment requires you to implement the application, not just to design one.

Assessment:

The assessment is as follows:

Documentation	25%
Code style/quality	25%
Justification of the strategy	25%
Operation of the interface	25%
