SeparationOfConcerns

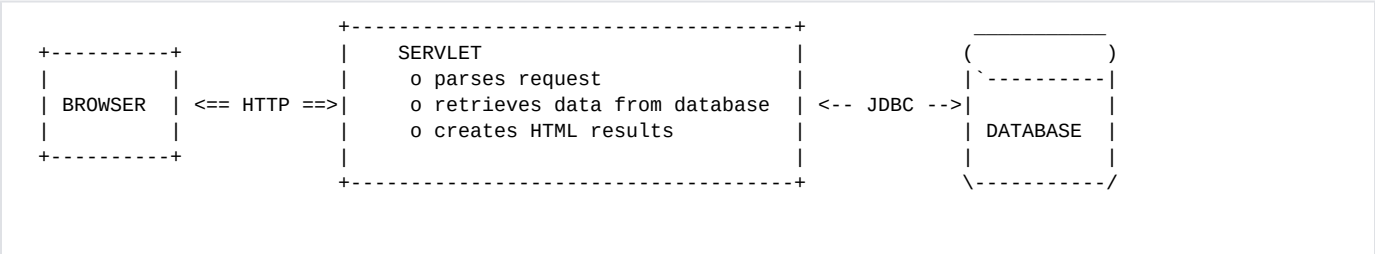
Separation Of Concerns

Separation Of Concerns (SoC) is a DesignPattern which states that problems can be seperated into a set of components each of which focus on only one core concern.

A Simple SoC Example

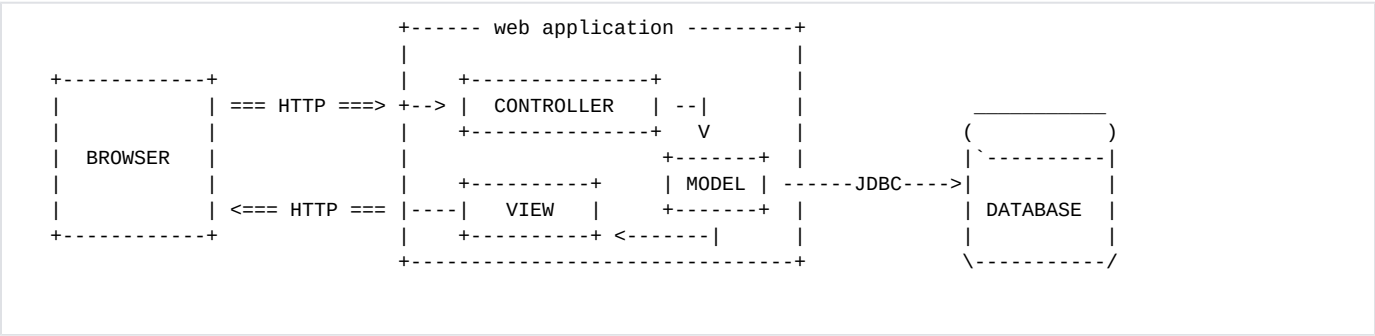
In order to understand SoC, let's look at a simple example: Java Servlet Programming.

When Java servlets first arrived on the scene, it wasn't too hard to find web application designed like this:



While for simple applications this design worked just find, it quickly became unmanagable as the problem scales. Any change to the HTML look and feel would require recompiling the servlet. Business logic and display logic was closely tied together. This is an example of no seperation of concerns. Everything jumbled together in one monolithic heap.

Thankfully designs and technology evolved and we soon saw a new design pattern being applied to Java web applications – Model View Controller (MVC). The MVC appraoch divides the single monolithic servlet into a Model (handles business data and rules), a View (handles presentation logic), and a Controller (parses request and controls execution flow). Various web frameworks built on this MVC idea such as Jakarta Turbine and Jakarta Stuts. So instead of the one servlet above, we end up with:



As we can see MVC is a type of Separation of Concerns. There are numerous advantages to this approach of software design, one of which is the easier application of InversionOfControl.

Additional Resources

If you'd like to learn more about SoC:

- Multi-Dimensional Separation of Concerns