**Distributed Counter Code Test**

Sharing resources between nodes is a challenge engineers have always faced when writing distributed systems. The difficultly arises when coordinating simultaneous access to the resource and ensuring that it is done fairly and without corrupting the resource.

Your task is to write a server and a client that allows multiple clients to simultaneously access a counter - the resource. Clients should be able to increment and decrement the counter and get its current value, via a protocol of your own design. The protocol should be over TCP/IP and not be built upon HTTP, RMI or similar. Care should be taken to persist the counter across restarts and/or failures, and a database should not be used. You may use whatever libraries or frameworks you deem fit to assist you in the task.

A command line interface is sufficient.

About the deliverable:

\*) it should consist of a tar.gz or zip file containing the source code, resources, libraries and any scripts needed to build the server and the client. make/ant/maven/gradle are acceptable choices for build tools.

\*) the server and the client should be written in Java, or a JVM based language e.g. Groovy, Scala

\*) a README explaining the reasons behind your decisions. Example topics would be: programming language, programming paradigms, libraries, frameworks, protocols, architecture, optimisations, coding style, test strategy and anything else you deem relevant. A few sentences on whatever topics you feel are relevant to your deliverable is fine.

\*) The specification is intentionally open to interpretation to allow you to show your creativity and thought process while still meeting the requirements

\*) Regular software engineering practices are expected e.g. well tested code, well refactored code