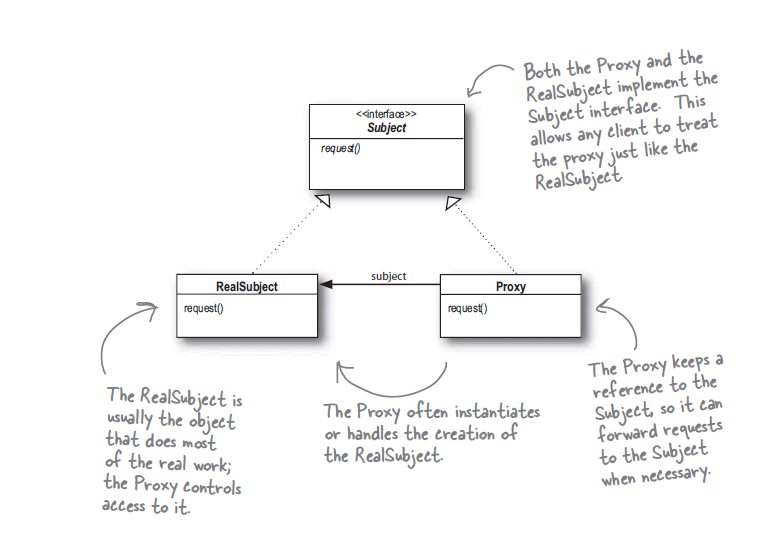
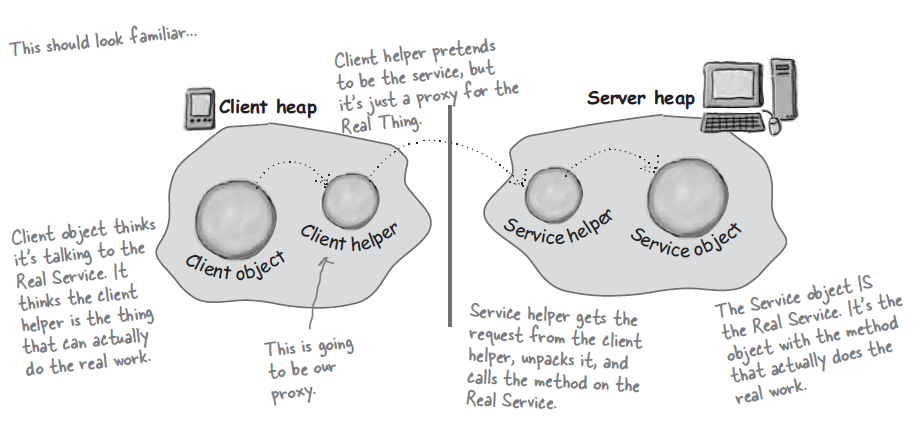
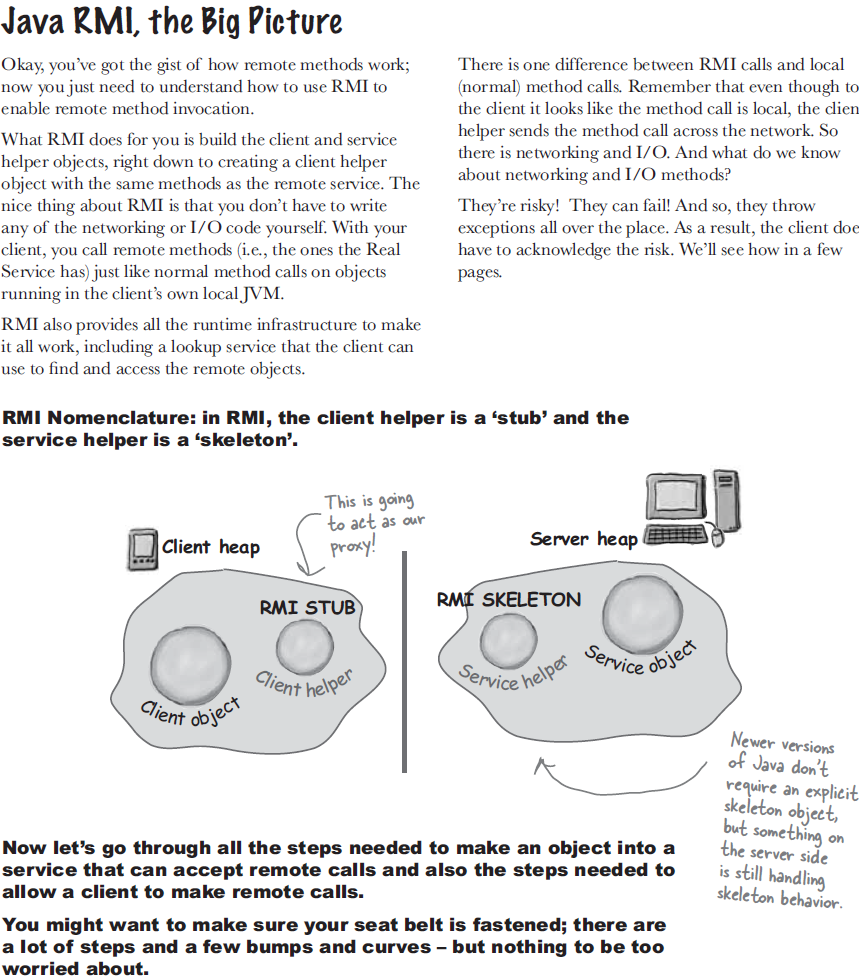
**The Proxy Pattern:**

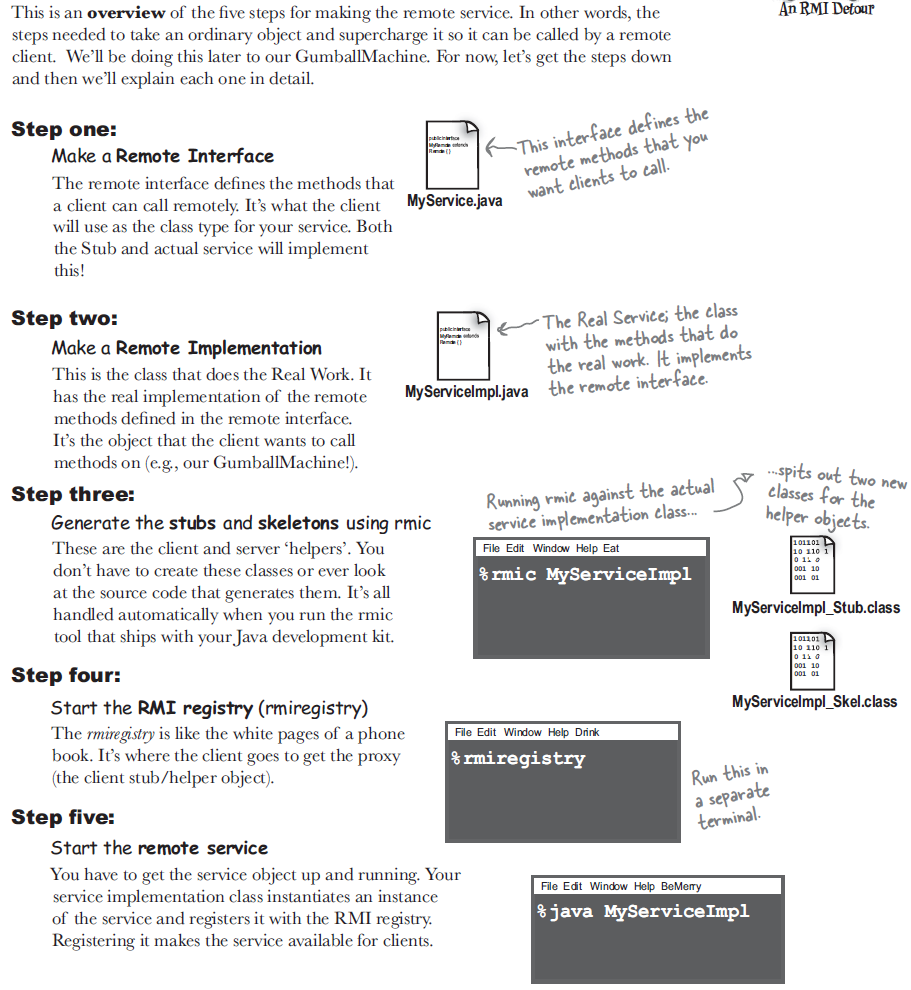
1. **Definition**: Proxy patterns create a representative object that controls access to another object which may be remote, expensive to create or needs security.
   1. There are 3 kinds of proxy: **Remote proxy, Virtual proxy** and **Protection proxy**
   2. Proxy and real subject both implements same interface such that proxy can replace real subject.
   3. Proxy instance composes of real subject instance on which it can delegate calls.
   4. **Class diagram**:

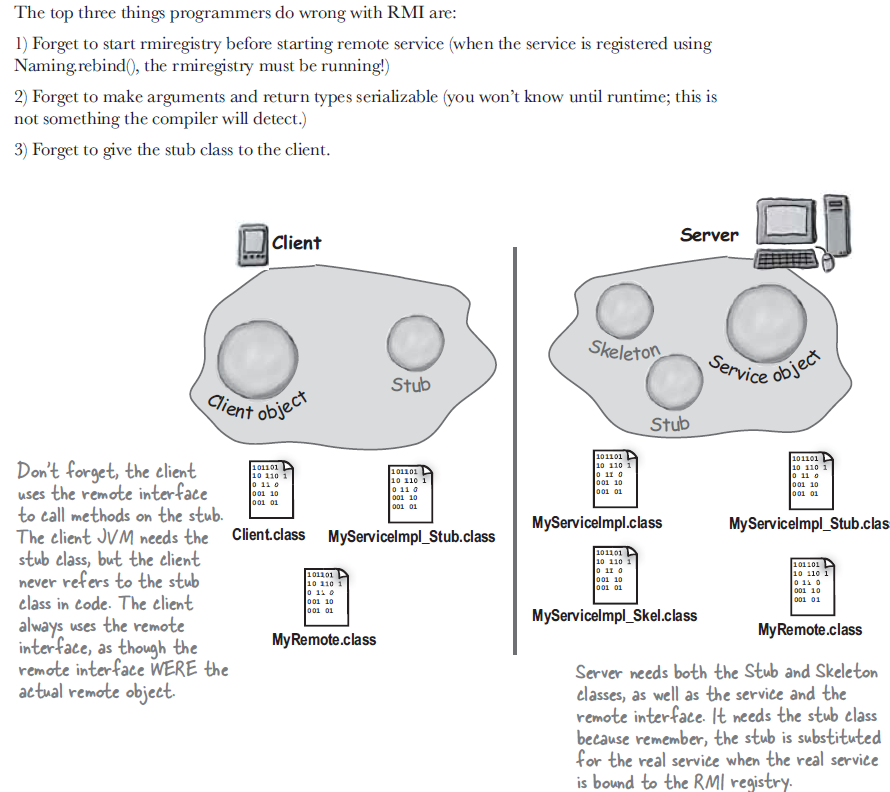


1. **Remote proxy**: Remote proxy acts as local representative to a remote object.
   1. Client object acts like it’s making a remote method call but what it’s actually doing is calling method on heap local proxy object that handles all lower level details of network communication.
2. RMI: RMI gives us a way to find objects on remote JVM and allows us to invoke their methods.

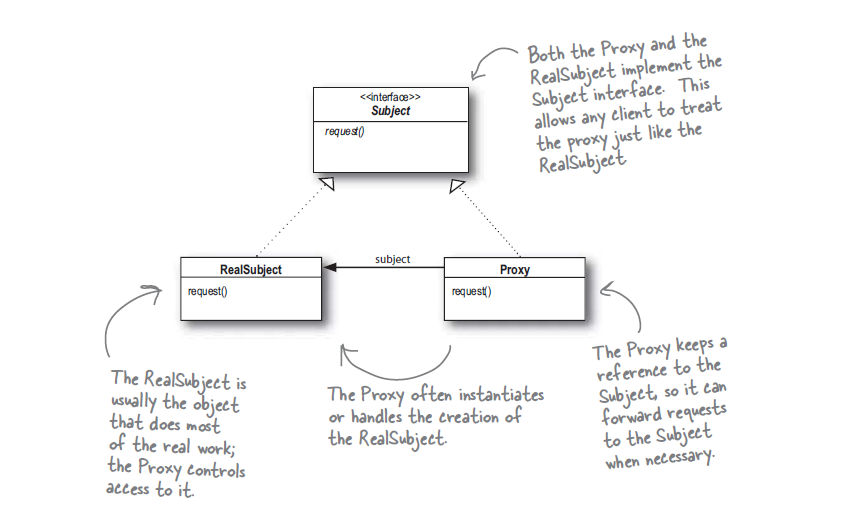








1. Definition: Proxy provide placeholder for another object to control access to it.
2. Class diagram:



1. **Dynamic proxy**: Java let us create proxy class on the fly which implements one or more subject interfaces and forward method invocations to subject we specify.

