1. When passing arrays in a remote procedure call, the array can be converted to a struct holding the number of the elements in the array. Both static and dynamic arrays can be passed this way, and converted back once received. Pointers would have to either be referenced by both IP and local memory address, or the objects could be copied to the callee's address space, and new pointers would be made to reference them. In this case, the programmer may have to worry about memory leaks, depending on how well the RPC is implemented, since creation and deletion of pointers could happen on either machine.

2. The client stub is eventually downloaded to the client to allow it to make remote calls from the server. The servers are first set up and registered in the RMI Registry. The client stub is held on the server until the client has requests and checks the codebase information on the server. After that, it can request the stub. Java arranges it this way for security, since the client has the opportunity to make sure all of the codebase information is good before downloading any potentially harmful code. Once it downloads the stub, it can start making requests, and it needs to make sure that the requests are safe.

3. Copy on reference is when a client attempts to read or reference something from the server, that item is copied to the client. Copy on write is where the client can read remotely, but if it attempts to write to the item, it is copied to the client's local machine. Copy on reference is very useful for cacheing commonly used files locally for faster access, but copy on write requires less overhead for passing messages and reading many locations.

4. In Java, security is implemented by using policy files and domains. Classes are separated into domains by trustworthiness, and each domain has a specific security policy. Classes that were downloaded from a remote location are typically in the most restricted domain, commonly known as a sandbox, local user-defined classes are less restricted, and built-in classes are the least restricted and have their own special domain. Each user has their own policy file which defines additional protocols for the system policy file. Java's Security Manager class handles the enforcement of these policies.