gmmq36

Part 1

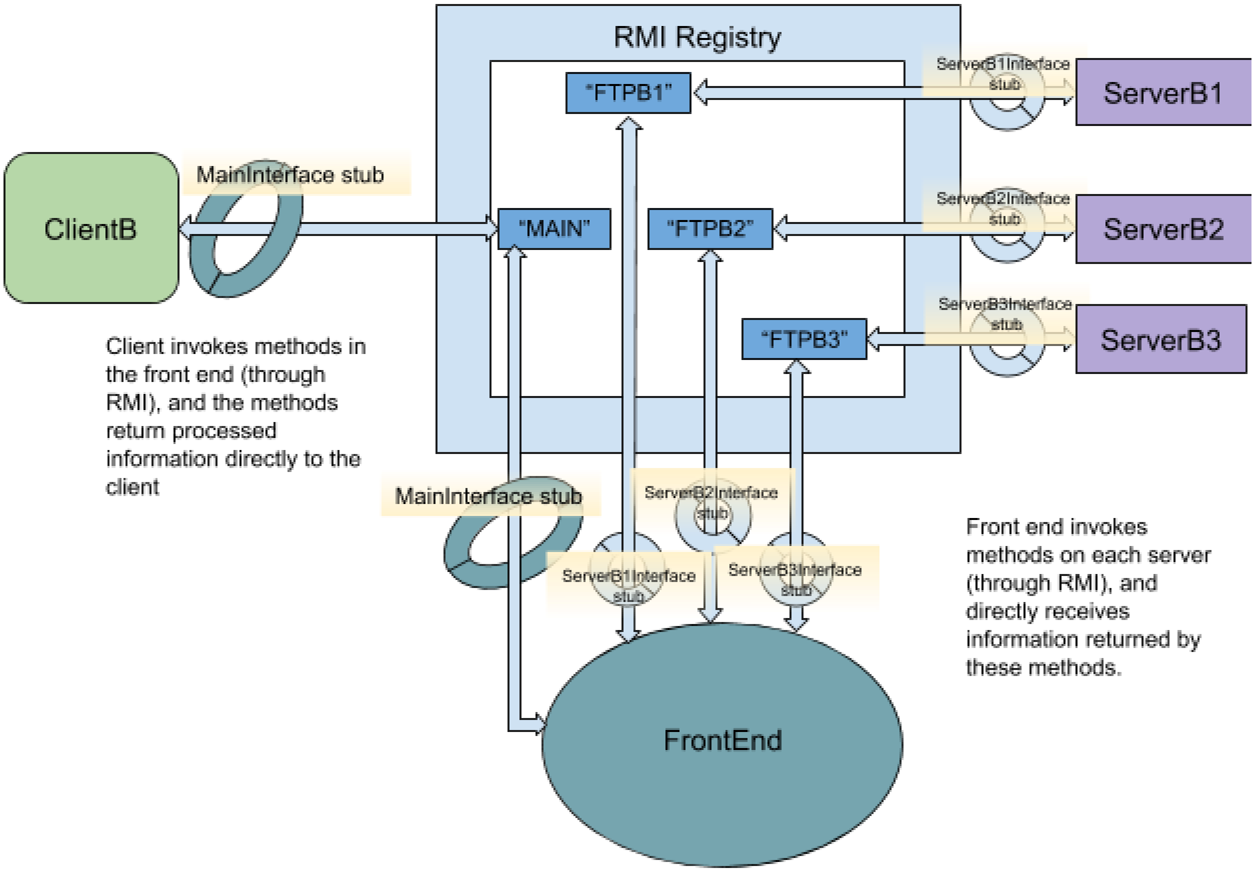
a) Why can RMI simplify the construction of distributed systems, comparing to using socket programming?

The use of Remote Method Invocation (compared to socket programming) can simplify the construction of distributed systems for a few reasons. For one, there is no need to design request-reply protocols (as is the case in socket programming). Request-reply protocols can easily become cumbersome and error-prone when implemented in distributed system consisting of many components. Another reason is that RMI clients can directly invoke server methods that or return complex types. In distributed systems, this allows for more straightforward abstraction. Socket programming cannot do this as well as socket programs can only communicate by passing values in a few predefined data types. RMI also generally enables a higher level of abstraction which more easily allows for transparencies necessary for the construction of distributed systems.

b) How does the front-end in a distributed system facilitate access transparency to data replica?

In a distributed system, the front-end facilitates access transparency in that it provides consistent external access to data despite potential disruptions to individual elements within the distributed system. As long as at least one copy of the data exists in the distributed system, the front end will show that there is one copy of the data in the system—even if there are multiple copies of the data. The front-end therefore conceals from the client the presence of data replica.

Part 2a



Client invokes methods in the front end (through RMI), and the methods return processed information directly to the client

Front end invokes methods on each server (through RMI), and directly receives information returned by these methods.