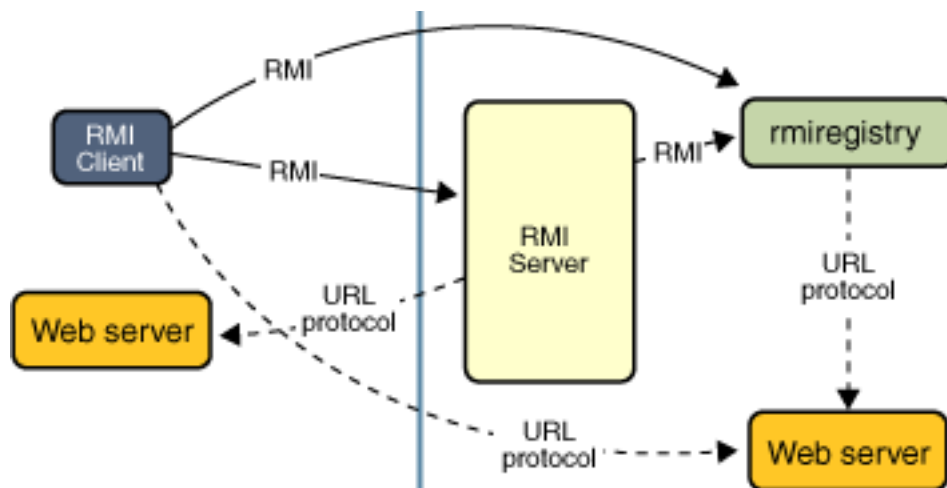


## Java RMI (Broker Pattern)

### 1. RMI [1]

RMI 애플리케이션은 클라이언트와 서버로 구성된다. 서버 프로그램은 remote objects를 생성하며 이러한 objects들에 대한 참조를 만들어서 클라이언트가 이러한 objects들에 대한 메소드를 호출할 수 있도록 한다.

### 2. RMI 구조 [1]



### 3. RMI 사용

코드 템플릿에 있는 RMI 예제는 먼저 RmiServer를 실행시킨 후, RmiClient를 실행하여 사용할 수 있으며, RmiClient에 있는 메인 함수는 다음과 같다.

```

import java.rmi.Naming;

public class RmiClient {
    public static void main(String args[]) throws Exception {
        RmiServerIntf server =
            (RmiServerIntf)Naming.lookup("//localhost/RmiServer");
        System.out.println(server.getMessage());
    }
}
  
```

#### 4. RMI 활용

---

- (1) 코드에서 서버 사이트에 Calculator 객체를 구현하고 두 개의 정수를 합하여 결과를 돌려주는 add 메소드를 추가하자. (주의! Calculator 객체는 그 객체를 위한 RmiCalculator 인터페이스를 구현하는데, 해당 인터페이스는 Remote를 상속해야 하고 RemoteException을 throws 해야 한다. 또한 RmiCalculator는 RmiServer와 유사하게 UnicastRemoteObject를 상속해야 하고, RmiCalculator를 위한 인터페이스를 구현해야 한다. 이에서 자동으로 나오는 오류는 오류 메시지의 추천을 통해서 해결한 후 실행하자)
- (2) 클라이언트에서 Calculator 객체를 불러 add 메소드를 실행시켜 결과를 출력하자.

#### 5. References

---

- [1] An Overview of RMI Applications, <https://docs.oracle.com/javase/tutorial/rmi/overview.html>