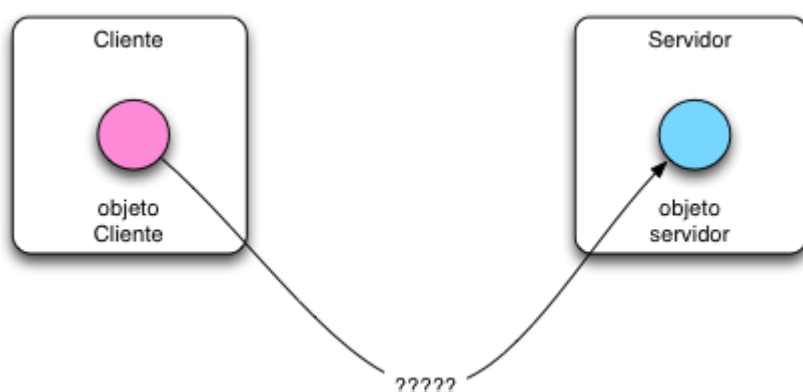


Tabla de Contenidos

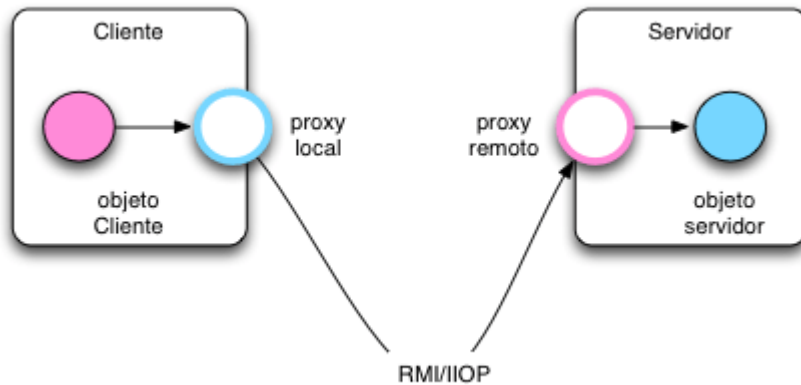


- Java RMI y limitaciones
- REST y SPA
- JSON-RPC
- JSON-RPC y Brutusin

JSON-RPC es un estándar basado en JSON para realizar invocaciones remotas. Ya hace muchos años que utilizamos Java RMI o RMI/IIOP para realizar peticiones remotas a objetos. El funcionamiento siempre es parecido, disponemos de un objeto que necesita comunicarse con otro ubicado en una máquina remota.



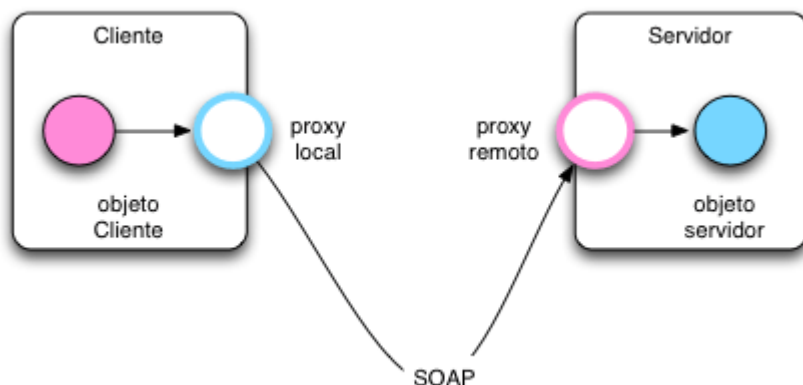
Lamentablemente al tratarse de máquinas diferentes no es posible un acceso directo entre los objetos. Esto se suele resolver con el diseño de **Proxys** clases que realizan la tarea de intermediario entre cliente y servidor añadiendo una capa adicional de comunicación.



Estos proxies simulan ser los objetos locales en cada máquina permitiendo al objeto cliente y servidor interactuar. El objeto cliente cree que tiene un objeto servidor real en su propia máquina, cuando realmente la petición se realiza a través de RMI hacia la máquina de destino.

Java RMI y limitaciones

Lamentablemente estas soluciones han tenido sus limitaciones ya que necesitaban estar desplegados cliente y servidor en una misma red ,ya que no se gestionaban las peticiones a través de HTTP. Para solventar esto aparecieron los servicios Web SOAP que permiten realizar llamadas entre objetos remotos utilizando HTTP .



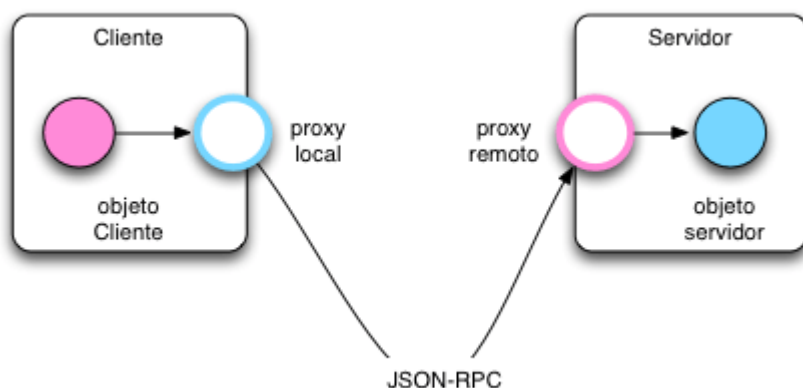
Esta fue en su momento una solución brillante , pero los años pasan y SOAP poco a poco ha dejado de estar tan de moda. Los servicios REST han comenzado a ocupar su lugar debido a que son mucho más sencillos de manejar por parte de cualquier cliente. Los mensajes SOAP están formados por XML y aunque un cliente puede invocarlos , la manipulación del mensaje a veces es complicada dependiendo de la plataforma. SOAP sigue teniendo su hueco ya que en algunos casos protocolos como WS-Security o WS-Atomic-transaction aportan ventajas sobre REST .

REST y SPA

Las arquitecturas basadas en REST han facilitado sobre manera el acceso a los datos desde arquitecturas SPA . Sin embargo hay que recordar que REST esta bastante orientado a operaciones CRUD y aunque HATEOAS nos permite ganar en flexibilidad no siempre es suficiente para todo el mundo. Hay en ocasiones que a la hora de definir interfaces de comunicación se queda corto.

JSON-RPC

JSON-RPC aporta un enfoque que vuelve a los principios de invocación remota de Objetos pero utilizando JSON como protocolo de comunicación. Uno de sus puntos fuertes es el soporte para mensajes complejos .



JSON-RPC no solo aporta una simplificación sobre SOAP , sino que además permite la ejecución tanto sobre HTTP como sobre WebSockets.

JSON-RPC y Brutusin

Brutusin-RPC es uno de los frameworks Java que se puede utilizar para construir microservicios basados en JSON-RPC. Como punto diferenciador, el framework hace uso de JSON-Schema como IDL (lenguaje de descripción de interfaz) lo que le permite dotar a los servicios de capacidades de autodescripción, facilitando enormemente su utilización y mantenimiento.

El framework ha sido creado por el español Ignacio del Valle Alles ([@idelvall](#)) de la empresa **DREAMgenics**, con los objetivos de potenciar la mantenibilidad de los servicios, hacer un uso correcto de la semántica HTTP, y proporcionar ciclos de desarrollo (codificación, test, depuración) rápidos y eficaces (basándose en la creación de microservicios), y está recibiendo una buena aceptación por parte de la comunidad opensource.