

REST v/s RPC

REST and RPC are two different architectures of communication between client and server. REST is mostly used as a REST API to interact with the same server from various client types like Mobile Apps, Web Apps, etc. On another side, RPC is widely used in services and microservices oriented architecture.

REST API

REST is an acronym of Representational state transfer and it is an architectural style that comply the constraints of REST. It has the following 06 constraints.

1- Client-Server

The client and server are two separate entities. Both can evolve and Scale independently

2- Stateless

Client-Server communication is stateless, which means each request from a client must contain all the data to complete itself on the server. As server does not remember the previous state of client communication.

3- Cache

Because of its Stateless nature, there are a lot of request/response cycles which is an overhead on the system. To reduce the overhead, APIs should be designed Cache friendly. The cache will be on the Client side.

4- Uniform Interface

There must be a standard way of communication between client-servers irrespective of technology or Application.

5- Layered System

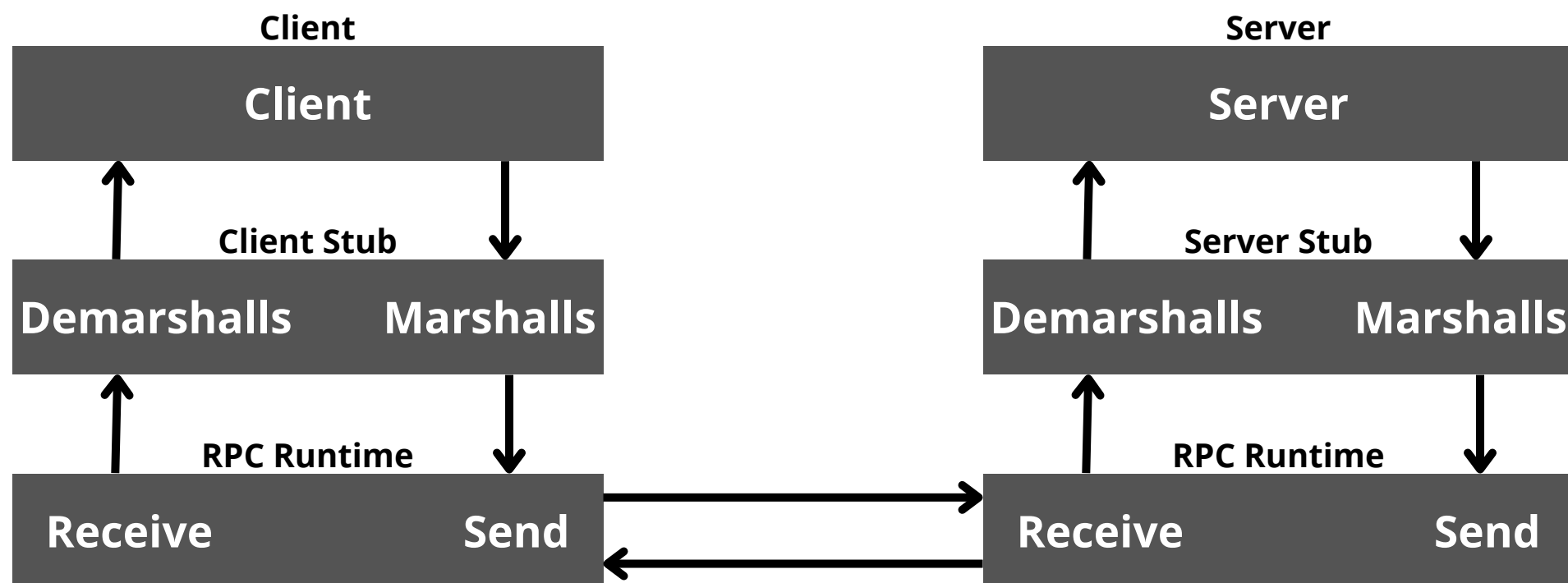
Applications are designed with multiple layers. Outer layers must not know each other, rather they know the intermediate layers. It will provide security and encapsulation between the systems.

6- Code On Demand

Code On Demand is the only optional constraint. A server can send the executable code to the client to enhance its functionality. It's a rarely used feature of REST.

RPC

A remote procedure call (RPC) is when a client invokes the function that resides on a remote server. It is similar to a function call you made on your local host.



- The client calls the client stub from in his own address space with the same signature as of function on the server.
- The stub pack(marshalls) the arguments in a message and send it to the server via the transport layer.
- The server unpacks (demarshalls) the arguments and calls the desired procedure.
- When the server completes the procedure, it returns the response to the server stub. The server stub marshalls the return value into the message and send it back to the client via the transport layer.
- The client-stub demarshalls the received message and forwards it to the client for execution.

REST

RPC

REST is resource-oriented.	RPC is action-oriented.
It supports HTTP methods POST, GET, PUT, PATCH, and DELETE	RPC only support POST and GET
It supports hypermedia and hyperlinks.	It does not support hypermedia and hyperlinks.
It allows a client to provide Content-types or accept headers.	It requires a payload of limited data types like XML.
It provides loose coupling between Client-Server.	It has tight coupling between Client-Server.
Support only request and response only.	Along with request and response, it also supports Bidirectional streaming.