Marshalling in gRPC

Marshalling is the process of converting in-memory data structures into a format suitable for transmission across a network, while **unmarshalling** reverses this process at the receiver.

In gRPC, marshalling is handled automatically using **Protocol Buffers (Protobuf)**. When the client invokes `SayHello`, the `HelloRequest` object is serialized into a compact binary format defined by the `.proto` schema. This binary representation is efficient, language-neutral, and platform-independent. The serialized message is then transmitted over HTTP/2 to the server.

Upon receiving the request, the gRPC runtime on the server **unmarshals** the binary data back into a `HelloRequest` object. The server processes the request and constructs a `HelloReply`, which is again marshalled into binary form and sent back to the client.

This approach provides several advantages. First, it significantly reduces message size compared to text-based formats like JSON or XML. Second, strict schemas ensure type safety and prevent ambiguity between communicating services. Third, because the schema is shared, different programming languages can interoperate seamlessly.

By abstracting marshalling away from developers, gRPC simplifies distributed communication while maintaining high performance and reliability—key requirements in modern distributed systems.