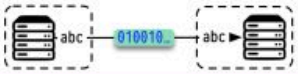
How to choose between RPC and RESTful?

RPC vs. RESTful — ByteByteGo.com

	RPC	RESTful
		
Coupling	Strong coupling	Weak coupling
Data format	Binary thrift, protobuf, Avro	Text XML, JSON
Communication protocol	TCP	HTTP/1.1, HTTP/2
Performance	High	Lower than RPC
Interface definition language (IDL)	thrift, protobuf	Swagger
Client code generation	Auto-generated stub	Auto-generated stub
Language framework	gRPC, thrift	SpringMVC, JAX-RS
Developer friendness	not human readable hard to debug	human readable easy to debug

Communication between different software systems can be established using either RPC (Remote Procedure Call) or RESTful (Representational State Transfer) protocols, which allow multiple systems to work together in distributed computing.

The two protocols differ mainly in their design philosophy. RPC enables calling remote procedures on a server as if they were local procedures, while RESTful applications are resource-based and interact with these resources via HTTP methods.

When choosing between RPC and RESTful, consider your application's needs. RPC might be a better fit if you require a more action-oriented approach with custom operations, while RESTful would be a better choice if you prefer a standardized, resource-based approach that utilizes HTTP methods.

Over to you: What are the best practices for versioning and ensuring backward compatibility of RPC and RESTful APIs?