gRPC

Taken from https://grpc.io

gRPC is a modern open source high performance Remote Procedure Call (RPC) framework that can run in any environment. It can efficiently connect services in and across data centers with pluggable support for load balancing, tracing, health checking and authentication. It is also applicable in last mile of distributed computing to connect devices, mobile applications and browsers to backend services.

The main usage scenarios

- Efficiently connecting polyglot services in microservices style architecture
- Connecting mobile devices, browser clients to backend services
- Generating efficient client libraries

Core features that make it awesome

- Idiomatic client libraries in 11 languages
- Highly efficient on wire and with a simple service definition framework
- Bi-directional streaming with http/2 based transport
- Pluggable auth, tracing, load balancing and health checking

The story behind gRPC

gRPC was initially created by Google, which has used a single general-purpose RPC infrastructure called Stubby to connect the large number of microservices running within and across its data centers for over a decade. In March 2015, Google decided to build the next version of Stubby and make it open source. The result was gRPC, which is now used in many organizations outside of Google to power use cases from microservices to the “last mile” of computing (mobile, web, and Internet of Things).

For more background on why we created gRPC, see the gRPC Motivation and Design Principles on the gRPC blog.

Documentation

Learn about key gRPC concepts, try a quick start, find tutorials and reference material for all supported languages and platforms:

⇒ **https://grpc.io/docs/**

gRPC in LabVIEW

See