gRPC:
A multi-platform RPC system

Louis Ryan

26th February 2016

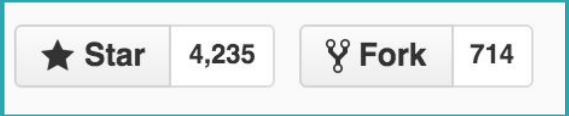
Google Cloud Platform

Microservices at Google

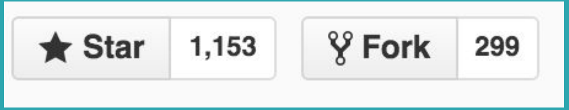
~O(10^{10}) RPCs per second.

Images by Connie Zhou

gRPC core


★ Star 4,235 Fork 714

gRPC java


★ Star 1,153 Fork 299

http://grpc.io

Open source on Github for C, C++, Java, Node.js, Python, Ruby, Go, C#, PHP, Objective-C

OVERVIEW

Google Cloud Platform

gRPC is ...

Open Source RPC framework that makes it ***easy*** to build a heterogenous distributed system.

- Free as in beer! (and licensing)
- Based on HTTP/2 today (multiplexed, works with the Internet)
- Payload agnostic (we've implemented proto)
- Streaming & Flow-Controlled
- Designed for harsh environments (timeout, lameducking, load-balancing, cancellation, …)
- Support in 10 languages & first class mobile support
- Layered & Pluggable - Bring your own monitoring, auth, naming, load balancing ...

Project Status

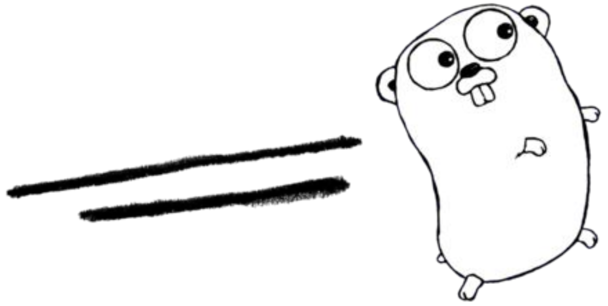
- Core features and protocol are fully specified
- Rolled out for public Google APIs and widely used internally
    - Lots of mobile adoption
- Approaching 1.0 (GA) release in all languages
    - Stable APIs for key features
- Benefit of layering on top of HTTP/2 standard
    - Interoperability with 3rd party proxies, tools, libraries..
    - WHATWG Fetch

Multiple Languages

C/C++

python™

Java™



C#

node js™

php

http://www.http2demo.io/

HTTP/1.1

HTTP/2

Protocol Buffers

IDL (Interface definition language)

Describe once and generate interfaces for any language.

Data Model

Structure of the request and response.

Wire Format

Binary format for network transmission.

```
message SubscribeRequest {
  string topic = 1;
}

message Event {
  string details = 1;
}

service Topics {
  rpc Subscribe(SubscribeRequest)
returns (stream Event);
}
```

Google Cloud Platform

Implementation Details

- Three complete stacks: C/C++, Java and Go.
- Other language implementations wrap C-Runtime libraries.
  - Hand-written wrappers to maintain language idioms
- Why wrap C?
  - Development costs & Implementation Consistency
  - Performance
  - Feature evolution
- Easy one line installation via packages e.g *npm install grpc*

USE CASES

Use Cases

Build distributed applications

- In data-centers
- In public/private cloud

Client-server communication

- Clients and servers across:
  - Mobile
  - Web
  - Cloud
- Also
  - Embedded systems, IoT

Access Google Cloud Services

- From GCP
- From Android and iOS devices
- From everywhere else

Images by Connie Zhou

HOW TO GET STARTED

Google Cloud Platform

Typical development workflow

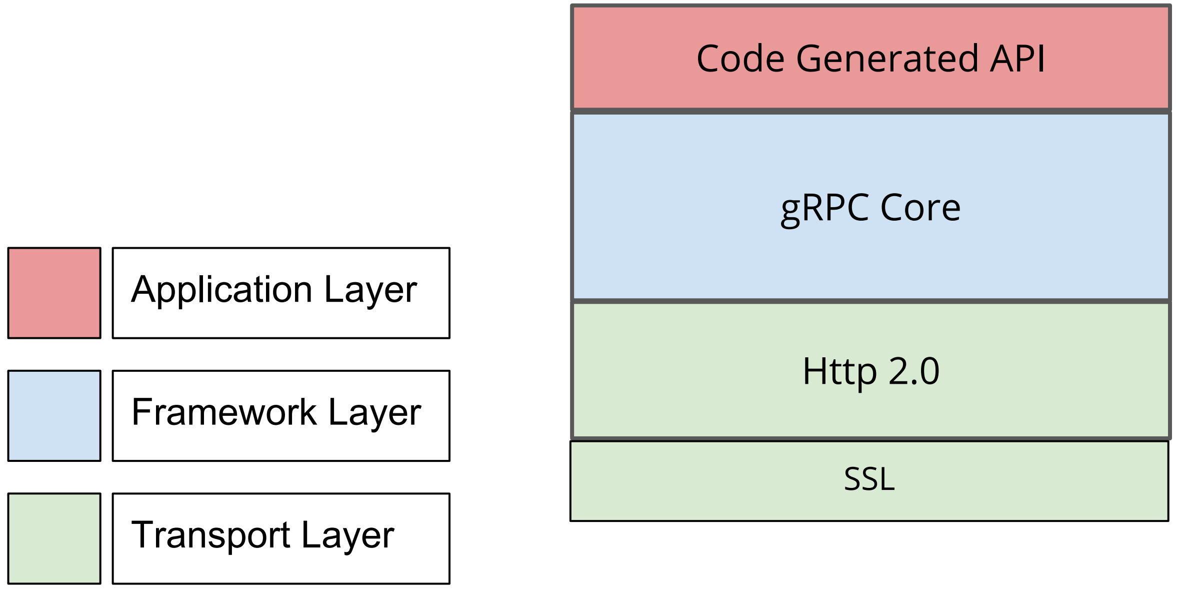
- Install
  - apt-get install protobuf-compiler
  - pip install grpcio
- Write the protos
- Use protoc to generate service interfaces, messages & stubs
- Implement services in server
- Client instantiates stub
- Test & Deploy

Advanced Deployment...

- Auth & Security - TLS [Mutual], Plugin auth mechanism (e.g. OAuth)
- Proxies - nghttp2, haproxy, Google LB, Nginx (in progress)
- Client-side load balancing - etcd, Zookeeper, Eureka, …
- Monitor & Trace - Zipkin, Google, DIY
- Mobile - Reconnect, QUIC
- Web - REST Adapter, WHATWG Fetch
- API Evolution - Protobuf, Versioning

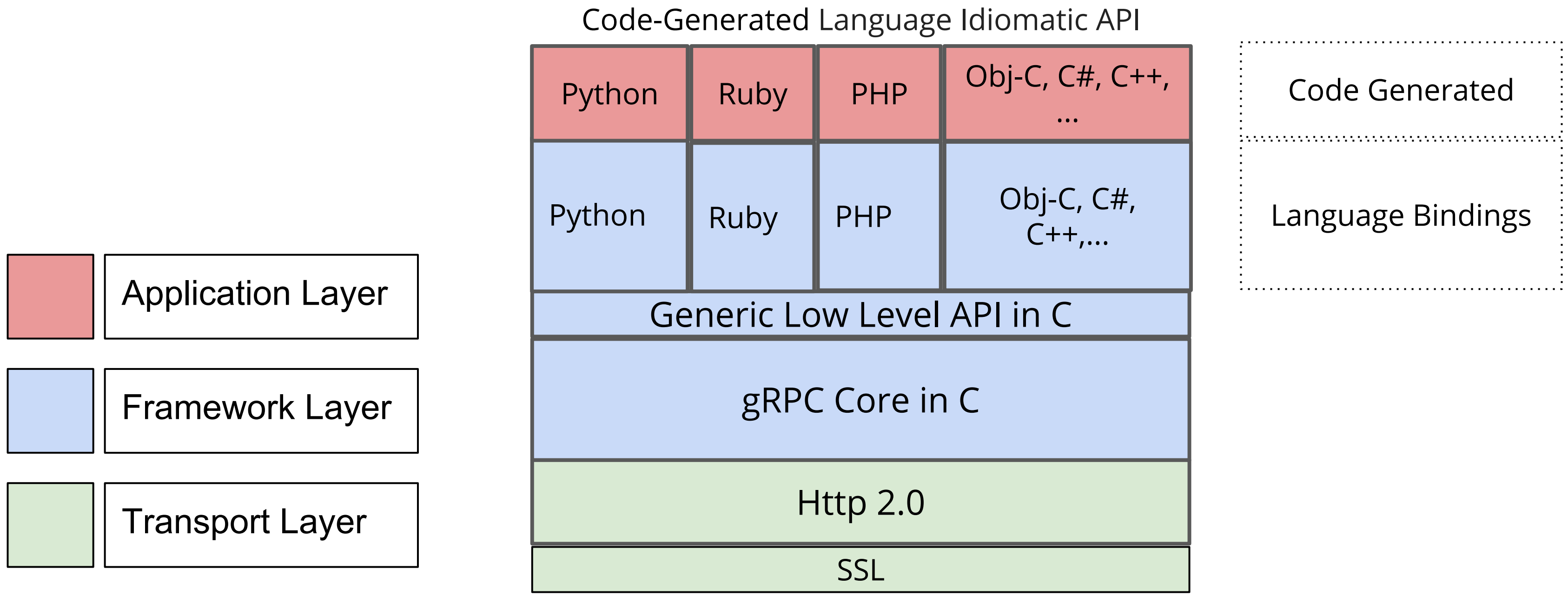
ARCHITECTURE

Architecture: Native Implementation in Language

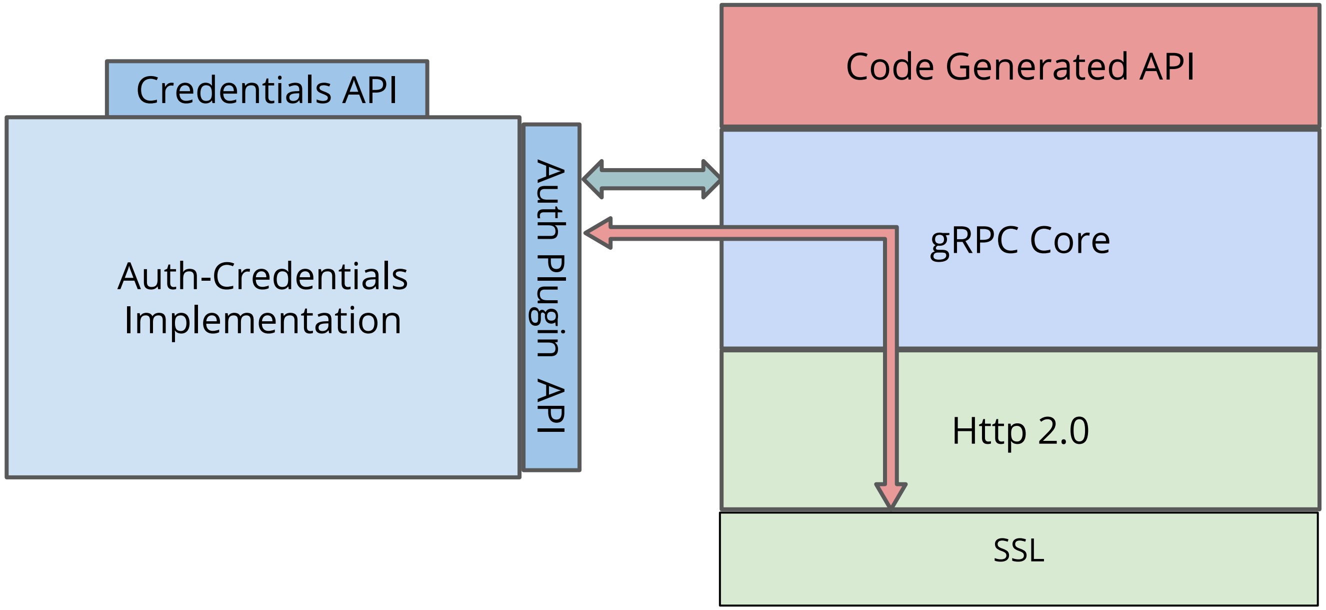


Planned in:
C/C++, Java, Go

Architecture: Derived Stack



Auth Architecture and API

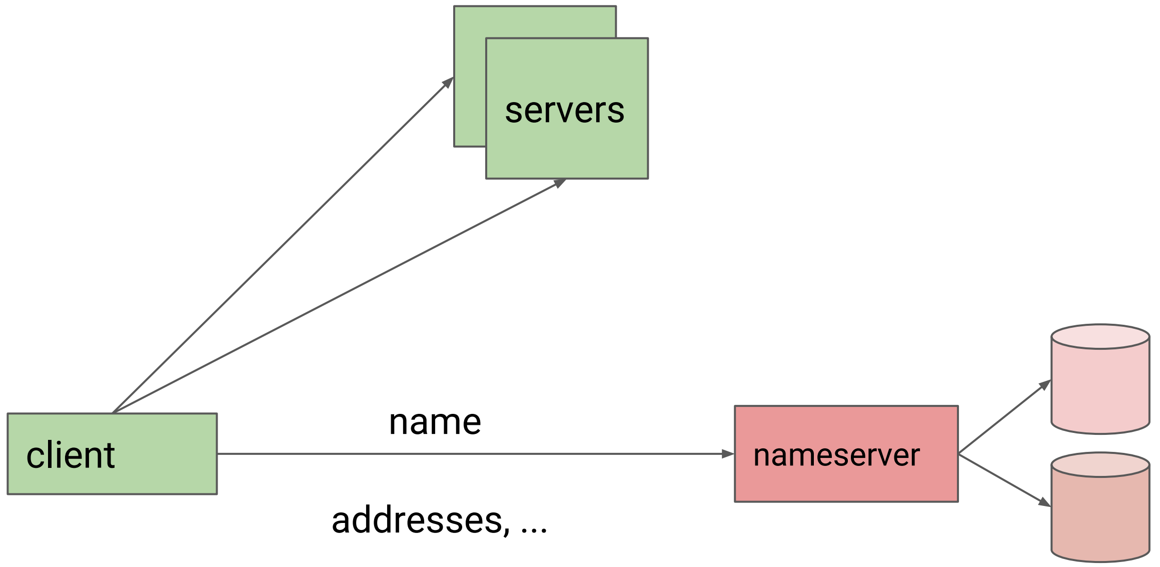


Metadata and Auth

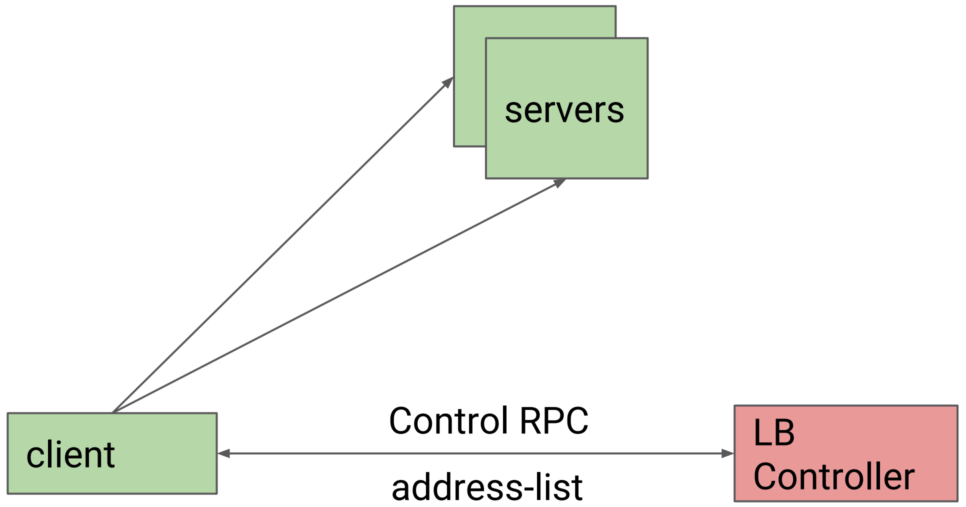
- Generic mechanism for attaching metadata to requests and responses
- Built into the gRPC protocol - always available
- Plugin API to attach “bearer tokens” to requests for Auth
  - OAuth2 access tokens
  - OIDC Id Tokens
- Session state for specific Auth mechanisms is encapsulated in an Auth-credentials object

ADVANCED FEATURES

gRPC: Naming

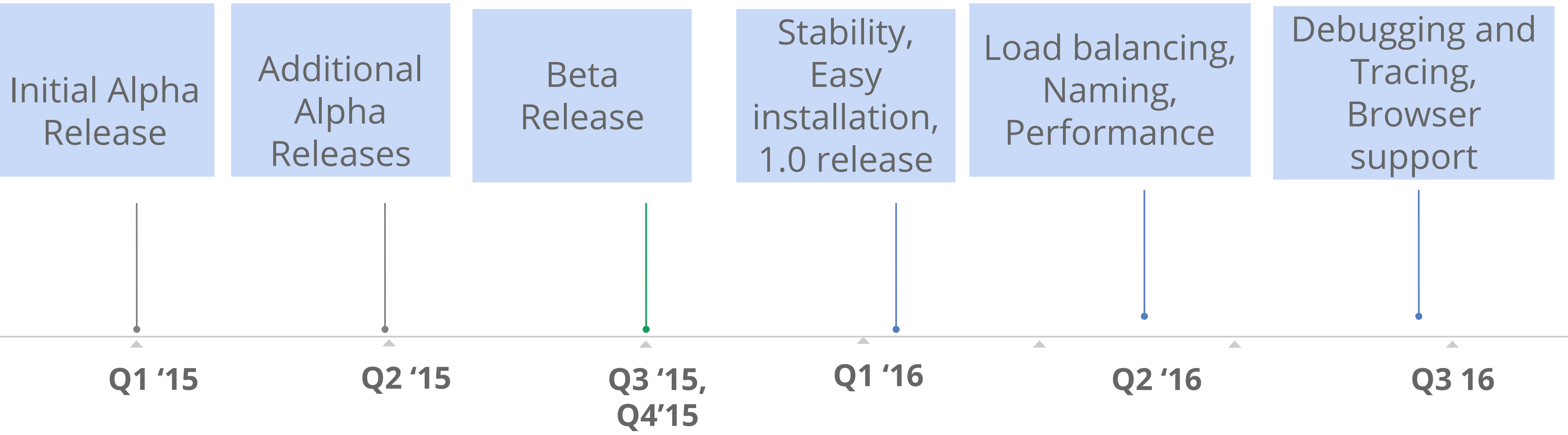


gRPC: LoadBalancing



ROADMAP..sorta !

Roadmap: Timeline


Initial Alpha Release
Additional Alpha Releases
Beta Release
Stability, Easy installation, 1.0 release
Load balancing, Naming, Performance
Debugging and Tracing, Browser support
Q1 '15
Q2 '15
Q3 '15, Q4'15
Q1 '16
Q2 '16
Q3 16

Thank you!

Twitter:	**@grpcio**
Site:	grpc.io
Group:	grpc-io@googlegroups.com