protobuf

Protocol Buffers

& gRPC

Because milliseconds matter

What will we be Learning Today?

Content

- ★ **Protocol Buffer**
  - ○ What is Protobuf?
  - ○ Why use Protobuf over JSON.
  - ○ Challenges
- ★ **gRPC**
  - ○ What is gRPC?
  - ○ Why use gRPC over REST.
  - ○ Types of gRPC APIs
  - ○ Ways to Implement gRPC in your Project.
  - ○ Challenges
- ★ **Recommended Use**
- ★ **Helpful Resources**

Protocol Buffers

- ★ Protocol buffer is a way of serializing structured data.
- ★ Google developed Protocol Buffers for use in their internal services.
- ★ Platform and Language agnostic.
- ★ Serialization/Deserialization API in multiple languages.
- ★ Open Source Project.

Why ProtoBuf Over JSON?

- ★ Schema is Followed
- ★ Backward Compatibility
- ★ Less Boilerplate Code
- ★ Validations and Extensibility
- ★ Easy Language Interoperability
- ★ Smaller Payload Size
- ★ Faster Serialization and Deserialization
- ★ Suitable for microservices architecture spanning multiple languages

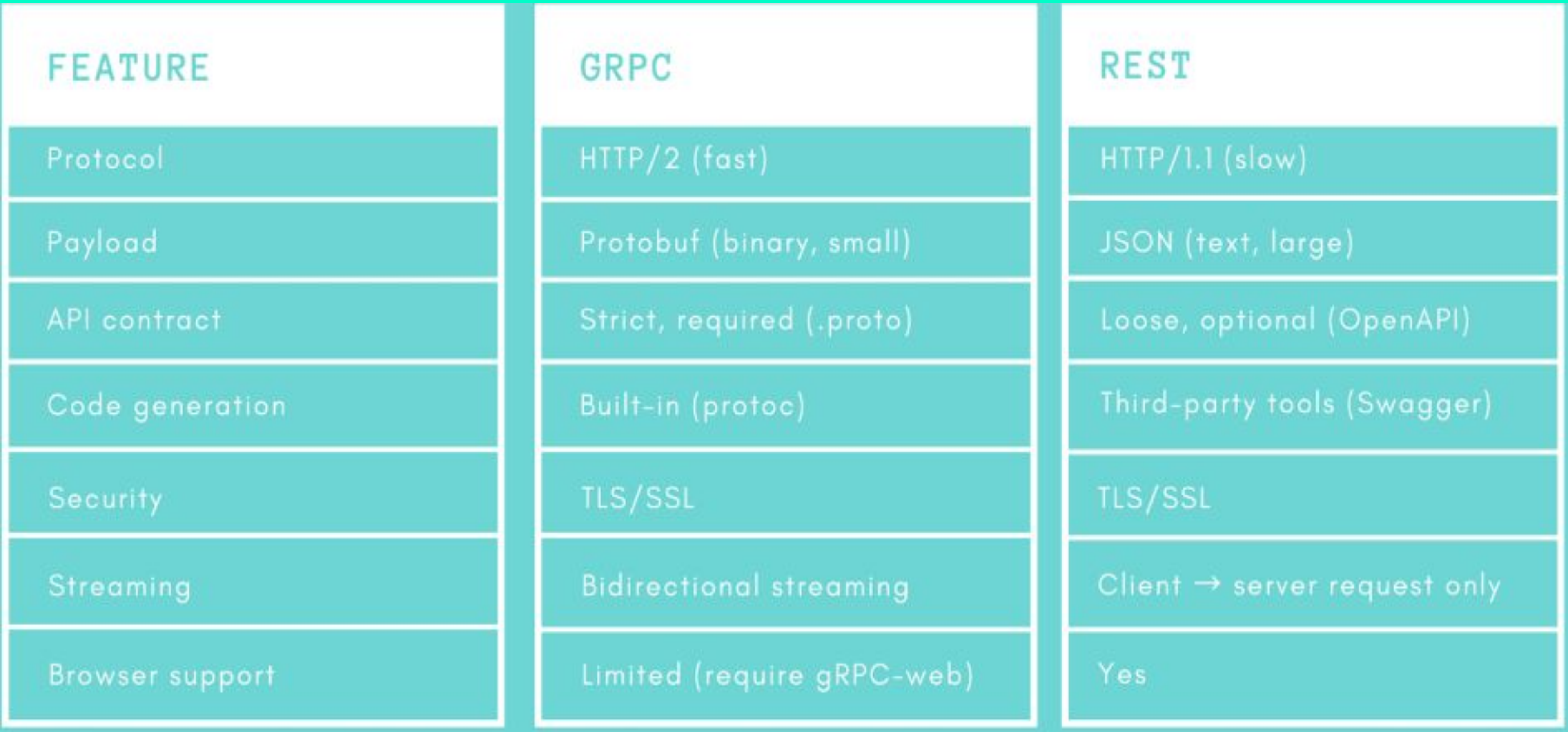
Challenges with Protobuf

- ★ Poor Readability
- ★ No Browser Support
- ★ Not recommended for largely Javascript Architecture
- ★ Looses advantage when Packet Size is large

gRPC

- gRPC stands for gRPC Remote Procedure Calls
- gRPC is a high-performance, open-source universal RPC framework.
- Simple and Idiomatic
- Performant and Scalable
- Interoperable and Extensible
- Used by big companies in production

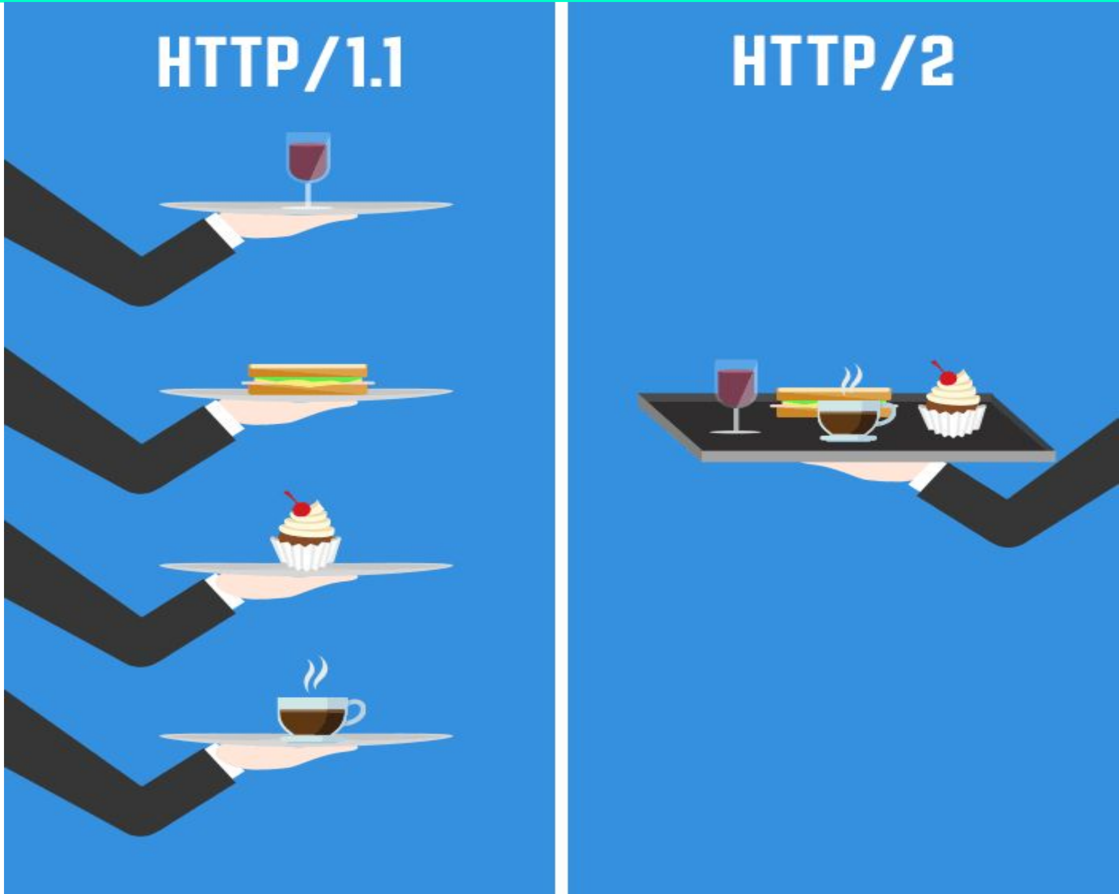
GRPC VS REST



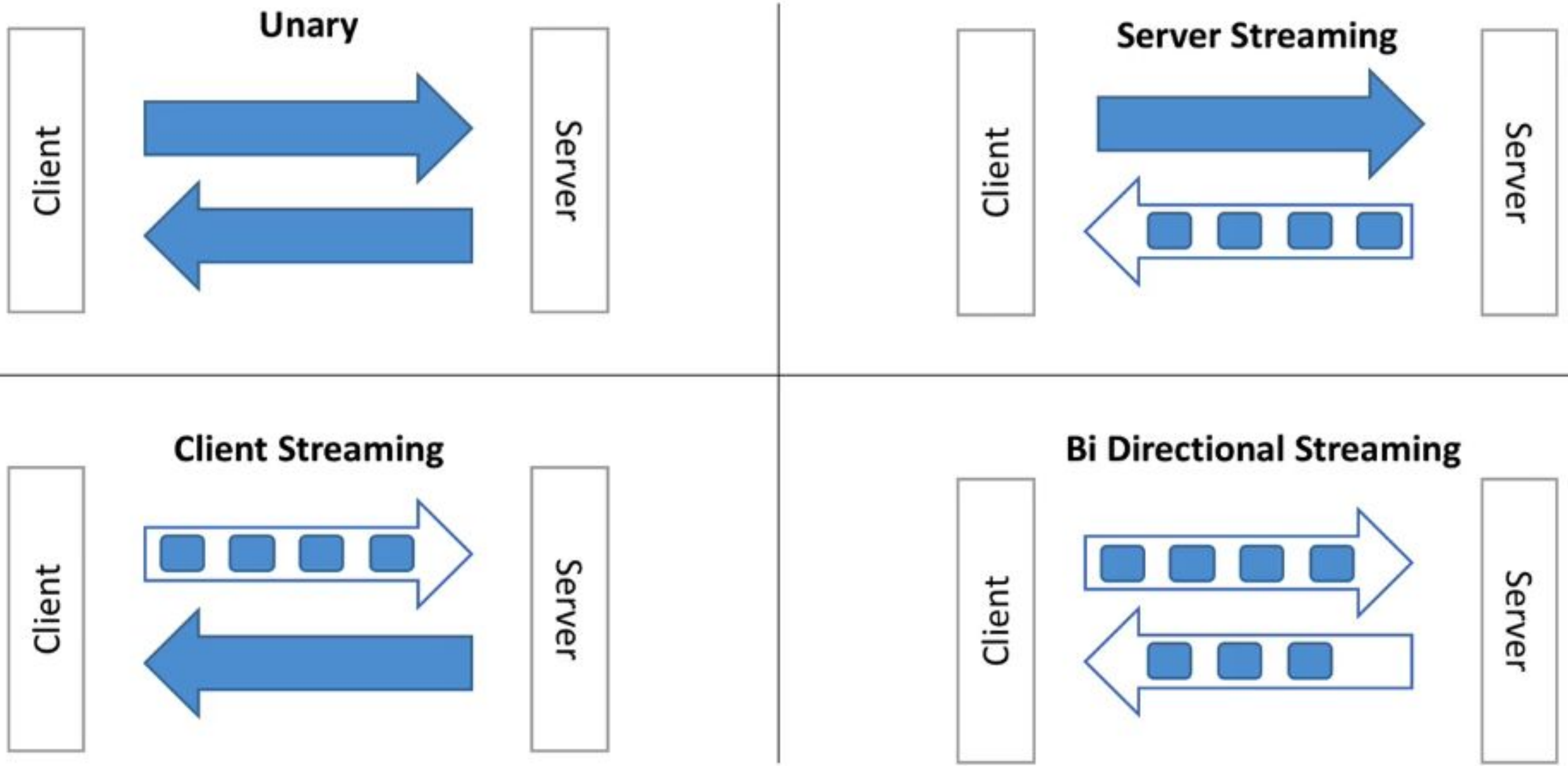
FEATURE	GRPC	REST
Protocol	HTTP/2 (fast)	HTTP/1.1 (slow)
Payload	Protobuf (binary, small)	JSON (text, large)
API contract	Strict, required (.proto)	Loose, optional (OpenAPI)
Code generation	Built-in (protoc)	Third-party tools (Swagger)
Security	TLS/SSL	TLS/SSL
Streaming	Bidirectional streaming	Client → server request only
Browser support	Limited (require gRPC-web)	Yes

Http/2 in One Slide

- ★ Single TCP Connection.
- ★ No Head-of-line blocking.
- ★ Binary Protocol.
- ★ Request -> Stream.
- ★ Header Compression
- ★ HTTP/2 Server Push



Types of gRPC Apis



Unary

Client

Server

Server Streaming

Client

Server

Client Streaming

Client

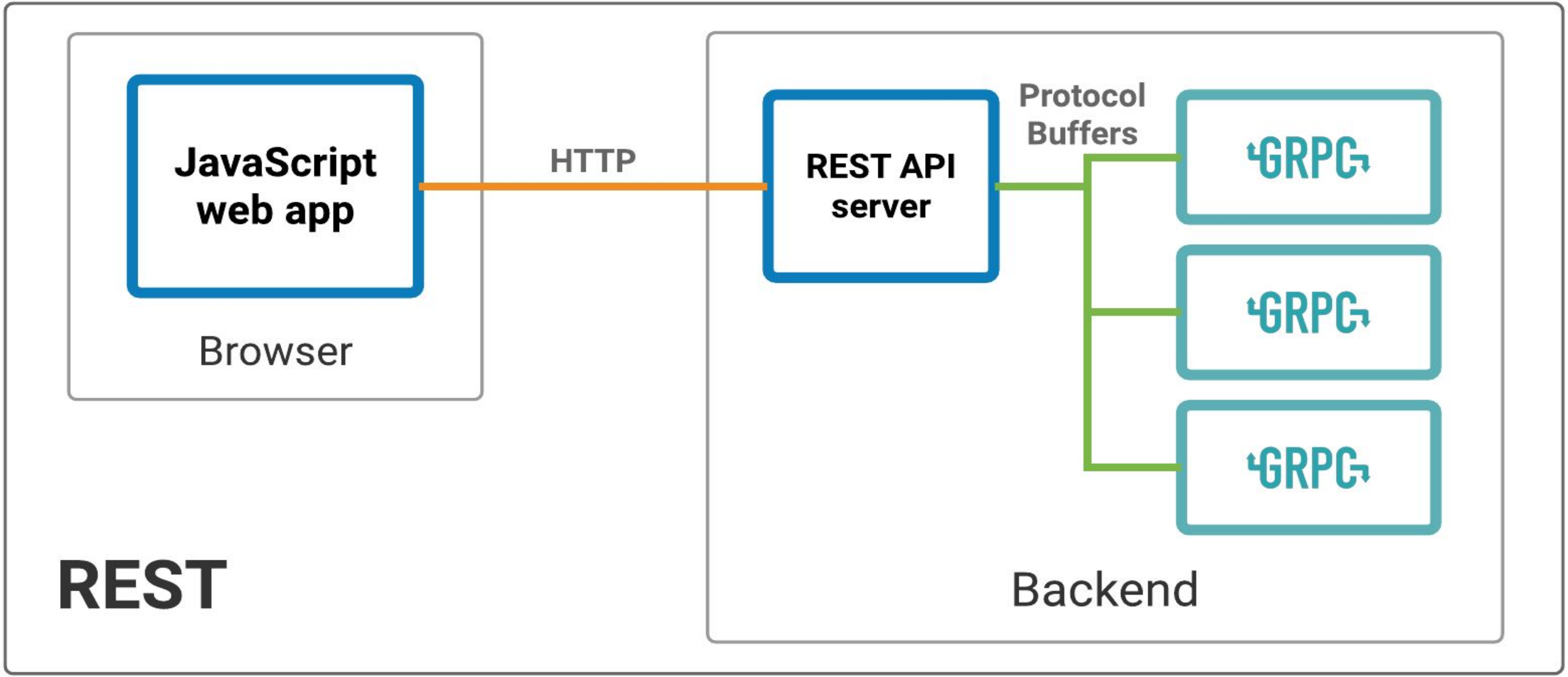
Server

Bi Directional Streaming

Client

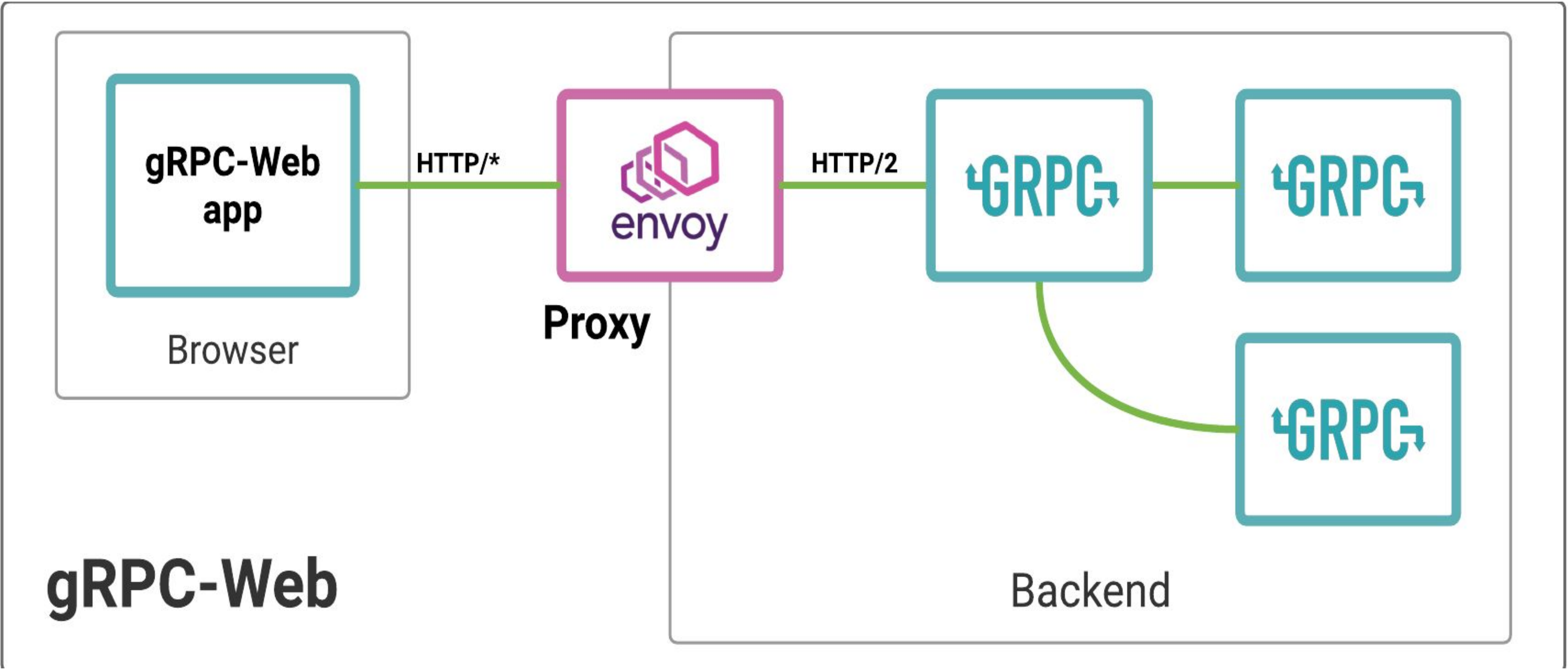
Server

Applying gRPC in Your Project



Source: https://grpc.io/blog/grpc-web-ga/

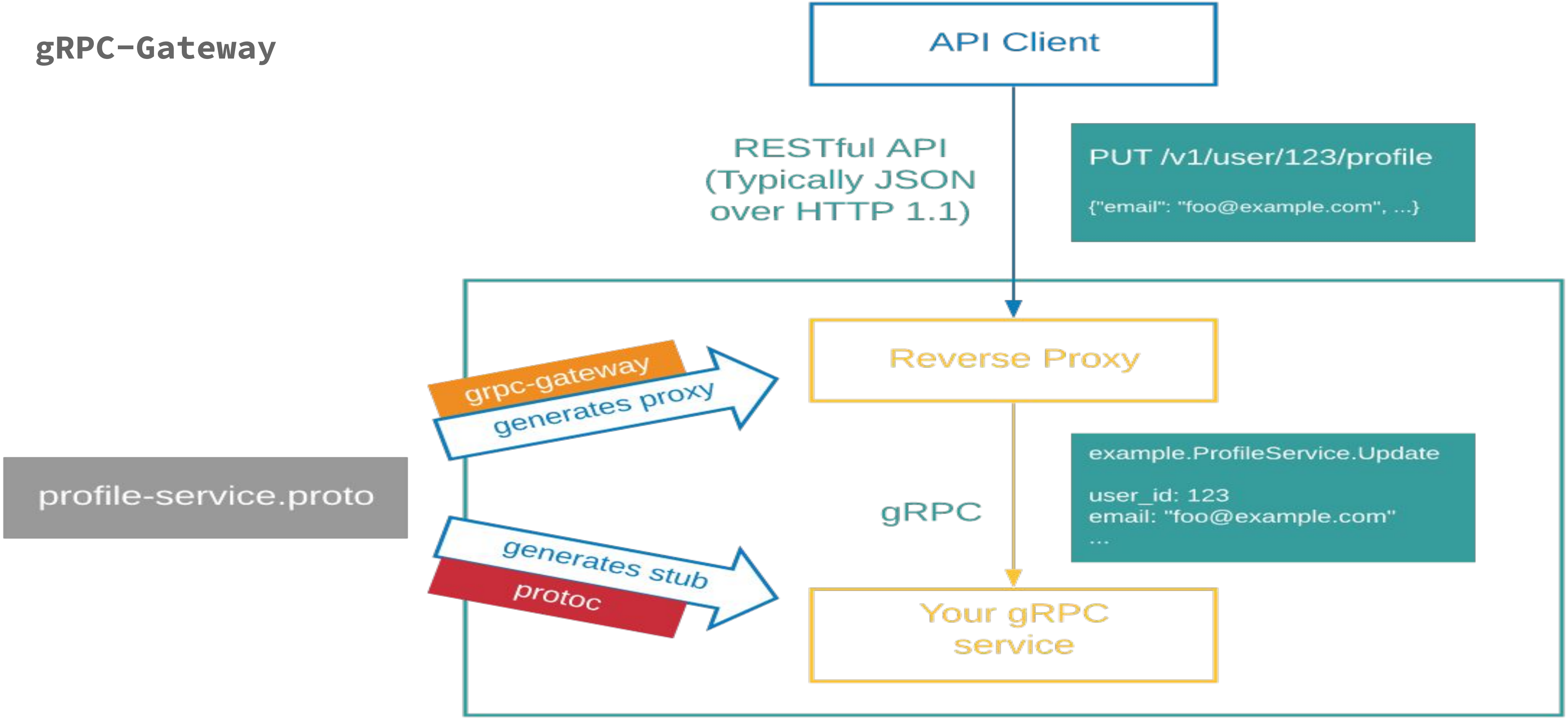
Continued...



Source: https://grpc.io/blog/grpc-web-ga/

Continued...

gRPC-Gateway



Challenges with gRPC

- ★ Limited Browser Support
- ★ Not Human Readable Format
- ★ Steep Learning Curve
- ★ Takes more implementation time than REST

Recommended Use

- ★ **Protocol Buffer**, apart from gRPC can be used in different cases like data exchange format between 2 services over message queue.
- ★ **Protobuf** serialization results in small message payloads, important in limited bandwidth scenarios like mobile apps.
- ★ **gRPC** can be used when services are internal and only communicate with other services.
- ★ **gRPC** can be used when services needs to stream high load of data within themselves.
- ★ **gRPC** is a good choice for Polygot environments.
- ★ **gRPC** is a great tool for mobile application backends.

Resources

- ★ **Protocol Buffers** : https://developers.google.com/protocol-buffers
- ★ **gRPC** : https://grpc.io/
- ★ **gRPC-Gateway** : https://grpc-ecosystem.github.io/grpc-gateway/
- ★ **gRPC-Web** : https://github.com/grpc/grpc-web, https://github.com/improbable-eng/grpc-web
- ★ **Awesome gRPC** : https://github.com/grpc-ecosystem/awesome-grpc
- ★ **Simple Suggestion** : https://www.bugsnag.com/blog/using-grpc-in-production https://www.trendmicro.com/en_us/research/20/h/how-unsecure-grpc-implementations-can-compromise-apis.html

Thanks!

Any Questions ?

Contact me:

Works At : Josh Software Pvt. Ltd. India

E-mail : sagar.sonwane@joshsoftware.com