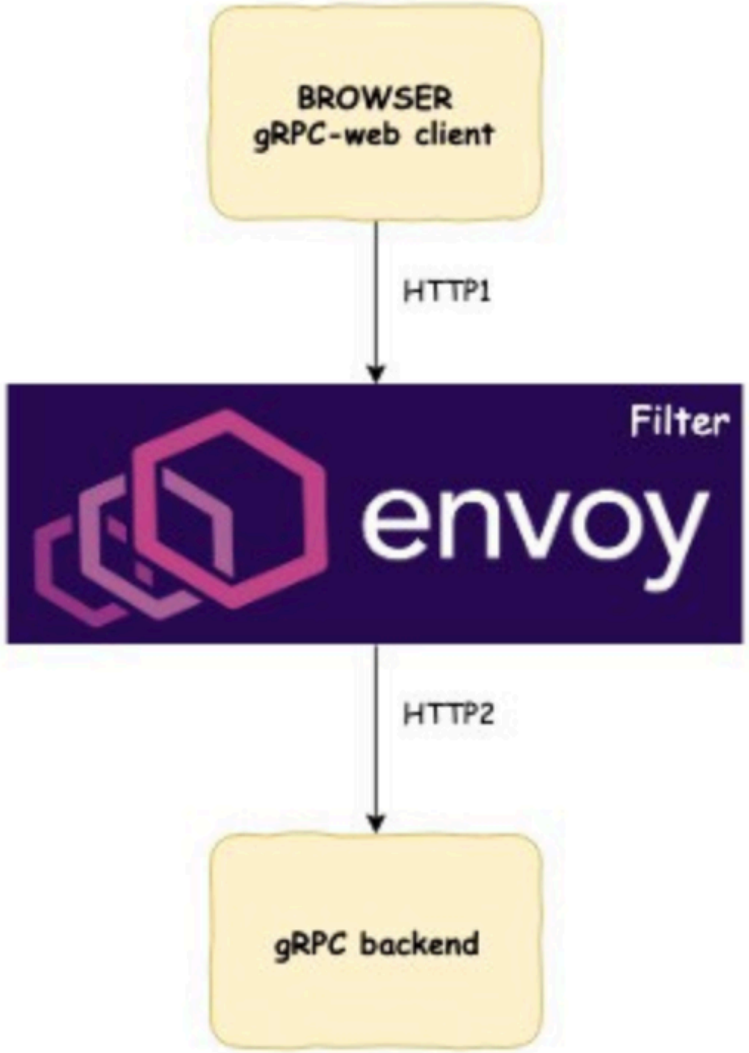
Reactive gRPC Chat App



Tech Stack

- FrontEnd: React
- BackEnd: gRPC server in Java
- Proxy: Envoy

Front-end + Proxy + Back-end

gRPC-Web

Client / Feature	Transport	Unary	Server-side streams	Client-side & bi-directional streaming
Improbable	Fetch/XHR	✓	✓	✗[19]
Google (`grpcwebtext`)	XHR	✓	✓	✗
Google (`grpcweb`)	XHR	✓	✗[20]	✗

- https://github.com/grpc/grpc-web
- gRPC web-client won't send HTTP2 requests. Instead, you need a proxy between your web-client and gRPC backend service for converting that HTTP1 request to HTTP2. gRPC web client has built-in support for Envoy as a proxy.
- As of now, client-side streaming is not supported.

The REST way

- gRPC-Web is a JavaScript client library that enables web applications to interact with backend gRPC services using Envoy instead of a custom HTTP server as an intermediary.
- It enables you to create *full end-to-end gRPC service architectures*, from the web client all the way down
  - Previously, if you wanted to use a gRPC-driven backend in conjunction with a web client you'd need to write REST API logic to translate HTTP calls to and from gRPC
    - grpc-gateway or grpc-json-transcoder
  - With gRPC-Web, client calls still need to be translated into gRPC-friendly calls, but that role is now filled by Envoy, which has built-in support for gRPC-Web and serves as its default service gateway.

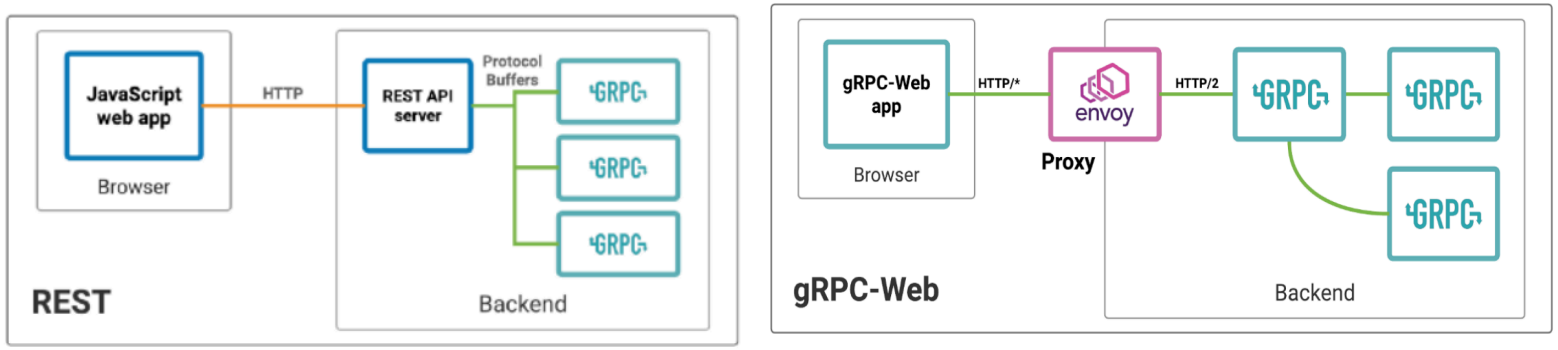


Figure 1. Client-backend interaction in a REST API vs. gRPC-Web

DeepDive

- Envoy config
- Protobuf Interface
  - Web Client (server-side streaming RPC)
  - Java Console Client (bidirectional streaming)