



KubeCon



CloudNativeCon

Europe 2020

Virtual

Design choices behind making gRPC available on Web Platforms

Wenbo Zhu, Google Inc.

Design choices behind gRPC-Web



KubeCon



CloudNativeCon

Europe 2020

Virtual

Outline

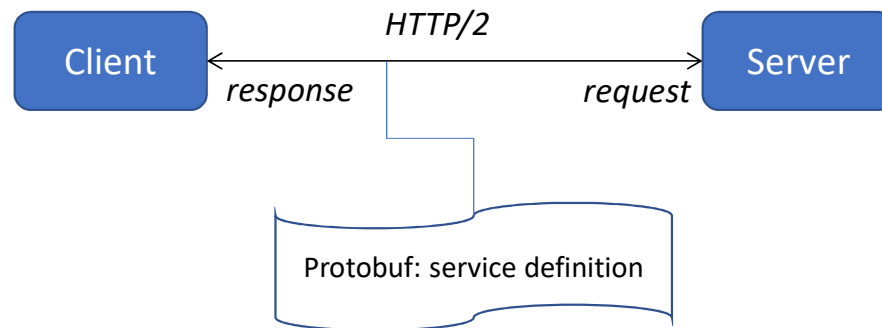
- Background: “RPC” on Web platforms
- Design choices: making the (right) tradeoffs
- Roadmap: a continued process

Q&A (15 minutes)

Background



gRPC in a nutshell



RPCs on Web platforms



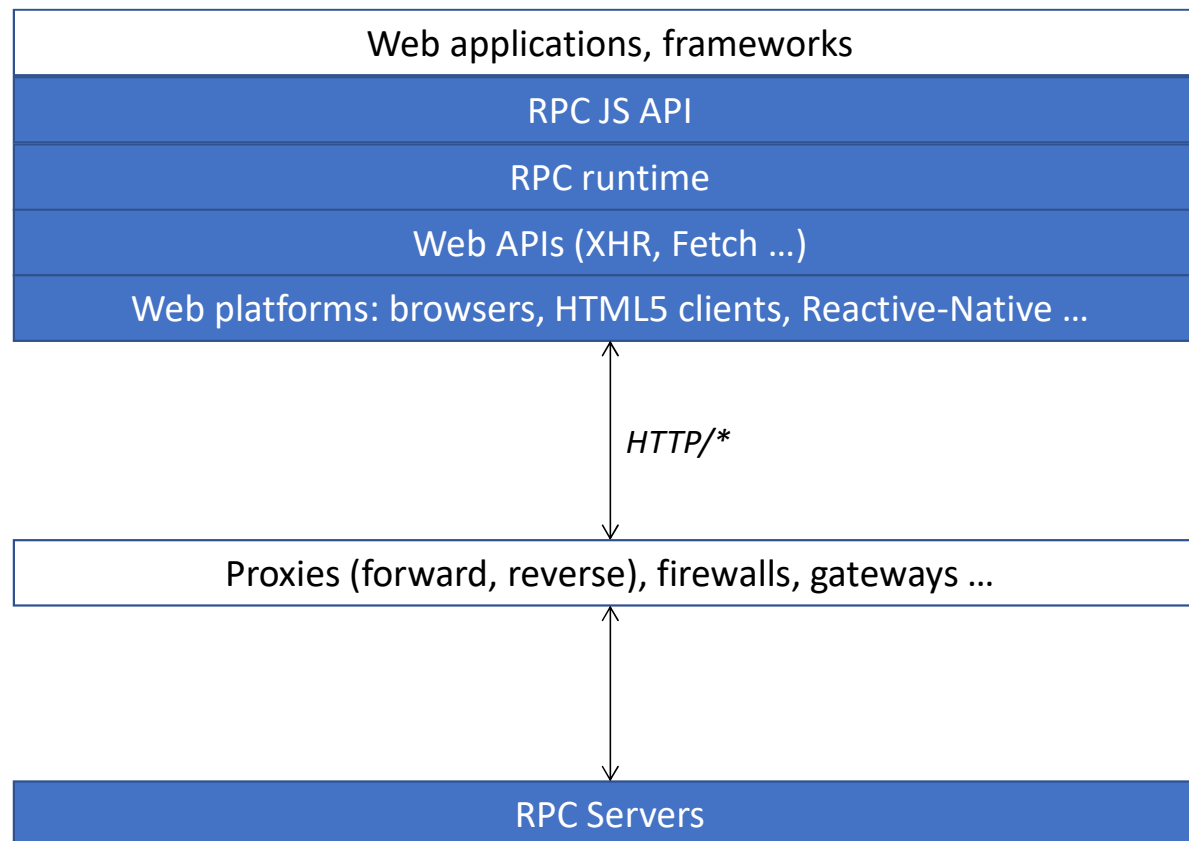
KubeCon



CloudNativeCon

Europe 2020

Virtual



(1) gRPC compatible



KubeCon



CloudNativeCon

Europe 2020

Virtual

Due to the Web platform constraints, a different protocol has to be designed, i.e. gRPC-Web.

gRPC-compatible

- Adopt the core grpc wire-protocol (h/2)
- Introduce only minimum protocol changes to enable gRPC on Web platforms
- Only extend the protocol to support Web-specific concepts such as CORS



be future-proof, and minimize server-side complexity

(2) Keep it simple



KubeCon



CloudNativeCon

Europe 2020

Virtual

Limit the streaming support to only server-streaming due to Web platform constraints.

KISS

- Avoid the complexity to support protocols that require fallback, e.g. websockets
- Unlike unary request-response or server-streaming, bidi-streaming or request-streaming is error-prone and less scalable (over Internet), i.e. RPC is not going to solve all your problems.

Web APIs are evolving, e.g. Fetch/streams.



don't invent things we may regret in future

(3) Works everywhere



KubeCon



CloudNativeCon

Europe 2020

Virtual

Now that we have made enough tradeoffs, we do strive to make the solution work everywhere.

Reachability

- Both browser and non-browser clients
- Older clients, e.g. IE 10
- Newer clients too, e.g. service workers

 wide adoption by Google's applications

(4) Focus on value-adds



KubeCon



CloudNativeCon

Europe 2020

Virtual

Prioritize features that advance the development experience.

Web development

- Focus on Web applications that interact with gRPC-based micro-services vs. gRPC-Web being a debugging tool
- APIs, code-gen & build, TS, Node ...

 improve gRPC adoption with a universal development experience based on protobuf

(5) Make REST a friend



KubeCon



CloudNativeCon

Europe 2020

Virtual

Yes, we want to benefit from the same infra that powers REST.

Web-compatible

- Server-side gateways rely on language-native Web frameworks to terminate gRPC-Web requests
- For protocol-agnostic features such as security, integrate, not reinvent.
- JSON support

 understand your target environments.

Current roadmap



KubeCon



CloudNativeCon

Europe 2020

Virtual

Bidi streaming

- Adopt fetch/streams to enable request-streaming and half-duplex bidi streaming [*]
- Publish a design guideline to support full-duplex streaming with dual HTTP requests

Gateways in more languages, in addition to Swift, .NET

Security features such as XSRF, XSS, CSP.

Protobuf improvements and performance.

[*] Chrome origin trial in progress for fetch/upload streams.

Community contributions



KubeCon



CloudNativeCon

Europe 2020

Virtual

Gateways: different languages, Envoy

Ecosystems: frameworks, different Web platforms

We want to hear your deployment experience

- <https://github.com/grpc/grpc-web>
- web@google.com

Q&A



KubeCon



CloudNativeCon

Europe 2020

Virtual



KubeCon



CloudNativeCon

Europe 2020



Virtual



KEEP CLOUD NATIVE

CONNECTED

