

 &



Tom Wilkie, GDG DevFest Ukraine, Oct 2017

[tom@kausal.co](mailto:tom@kausal.co) @tom\_wilkie

[github.com/tomwilkie](https://github.com/tomwilkie)





- Founder Kausal, “transforming observability”
- Prometheus developer
- Home brewer

Previously:

- Worked on Kubernetes & Prometheus at Weaveworks
- SRE for Google Analytics
- Founder/CTO at Acunu, worked on Cassandra





in gRPC @ Weaveworks

Tom

https://www.slideshare.net/weaveworks/grpc-weaveworks

Apps Recipes F1 Programming TeeVee House Home Brew Quick INIT The Best Independ...

SlideShare | Search Upload

Home 1 person clipped this slide

gRPC  
@ Weaveworks

Tom Wilkie, 14/03/2017  
Weave Online Users Group



## **Introduction**

What is gRPC? What makes it different?

## **Example**

Hello World!

## **Scratching the surface...**

Load balancing; Error propagation; Instrumentation...

## **Advanced Topics**

Distributed tracing; gogoproobuf...



# Introduction



```

type request struct {
    Name string `json:"name"`
}

type response struct {
    Message string `json:"message"`
}

func server(addr string) {
    http.Handle("/greeter", http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
        var req request
        if err := json.NewDecoder(r.Body).Decode(&req); err != nil {
            http.Error(w, err.Error(), http.StatusBadRequest)
            return
        }

        res := response{
            Message: "Hello " + req.Name,
        }

        if err := json.NewEncoder(w).Encode(&res); err != nil {
            http.Error(w, err.Error(), http.StatusInternalServerError)
            return
        }
    }))
}

```

```

func client(address, name string) {
    req := request{
        Name: name,
    }

    var buf bytes.Buffer
    if err := json.NewEncoder(&buf).Encode(req); err != nil {
        log.Fatalf("encoding error: %v", err)
    }

    resp, err := http.Post(address, "application/json", &buf)
    if err != nil {
        log.Fatalf("failed to post: %v", err)
    }
    defer resp.Body.Close()

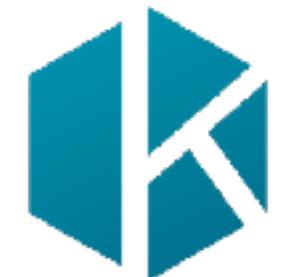
    if resp.StatusCode/100 != 2 {
        log.Fatalf("request error: %d", resp.StatusCode)
    }

    var res response
    if err := json.NewDecoder(resp.Body).Decode(&res); err != nil {
        log.Fatalf("decoding error: %v", err)
    }
}

```

Have you ever had to write code like this?

gRPC is here to save you...



```
syntax = "proto3";
package hello;
option go_package = "main";

service Greeter {
    rpc SayHello (HelloRequest) returns (HelloReply) {}
}

message HelloRequest {
    string name = 1;
}

message HelloReply {
    string message = 1;
}
```

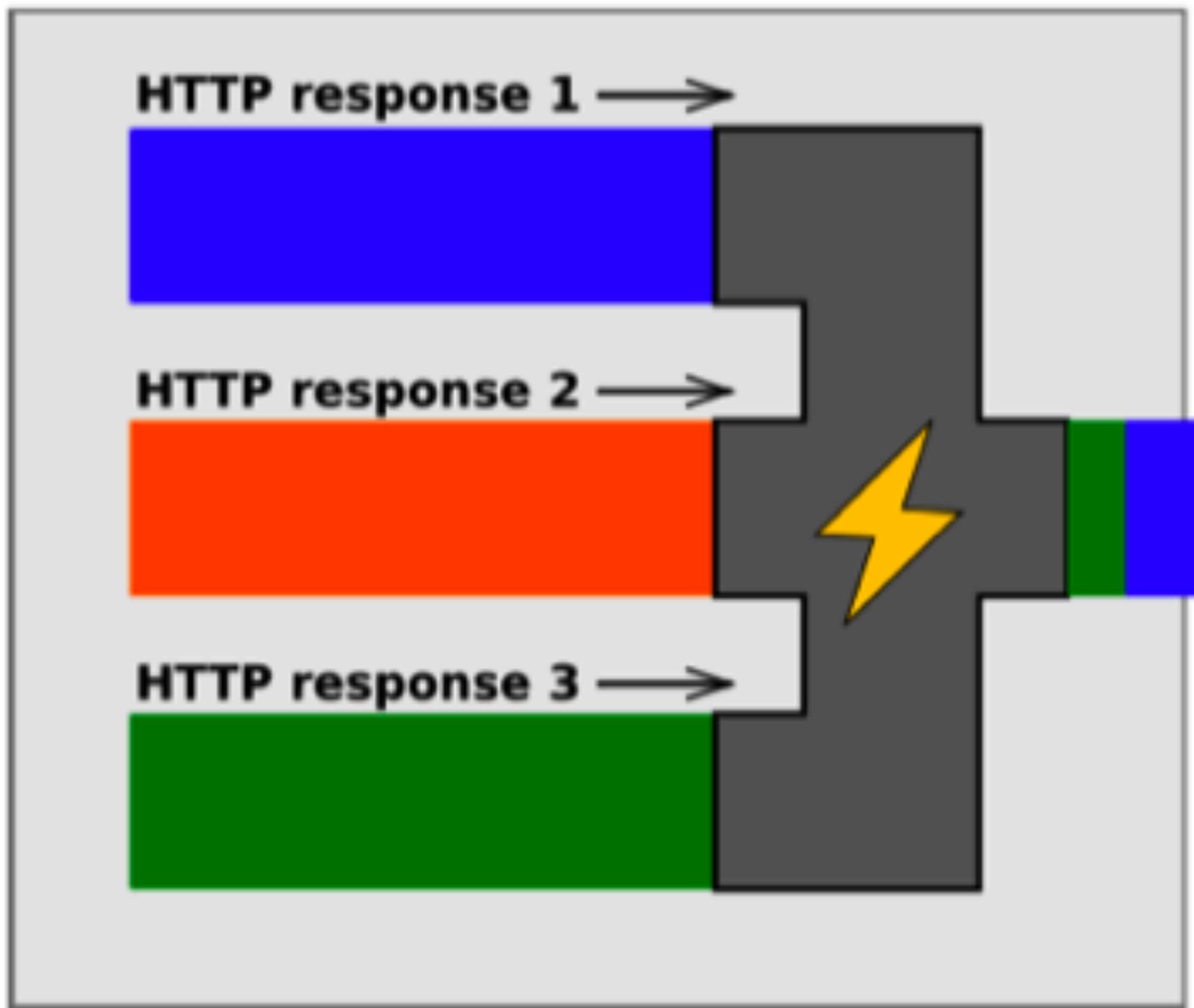


- 1991: CORBA
- 1997: Java RMI
- 1999: SOAP & WSDL
- 2000: REST HTTP/JSON
- 2008: Protocol Buffers
- 2009: Thrift
- 2015: gRPC



Everything Old Is New Again

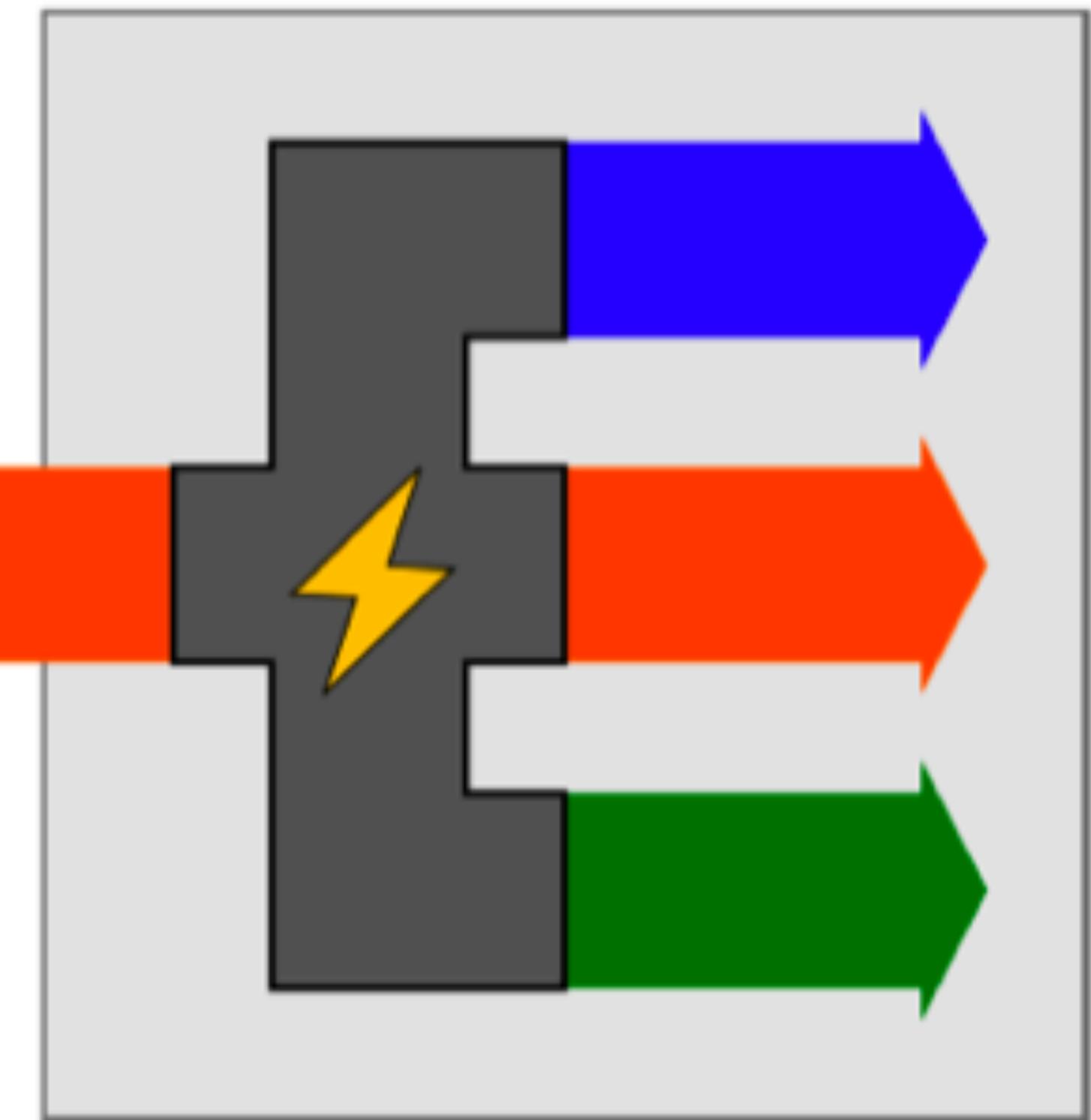
# Server



# Client

HTTP/2 →

Single TCP connection



<https://www.nginx.com/blog/7-tips-for-faster-http2-performance/>



<http://http2demo.io>

HTTP/2

C/C++ C#



Objective-C



Idiomatic support for 10 languages





NETFLIX



Google



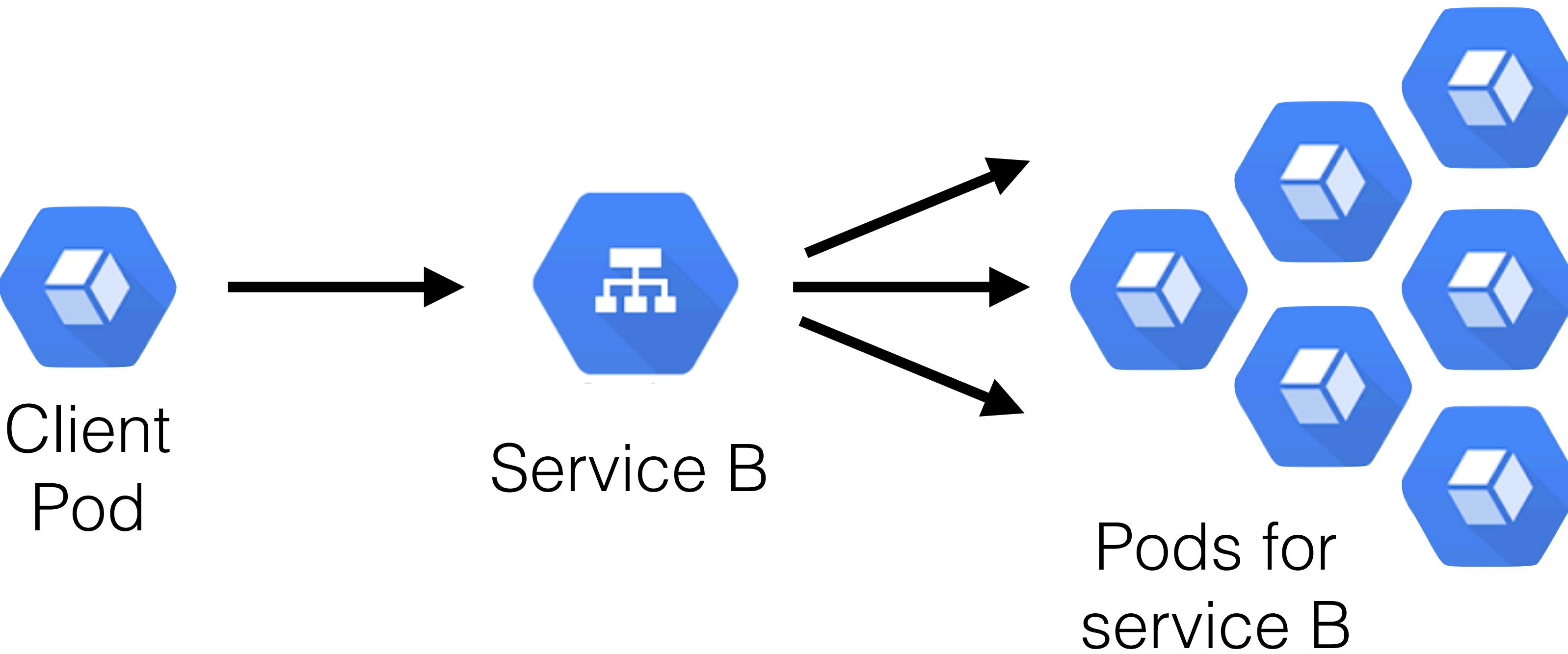
Who is using gRPC?

# Example



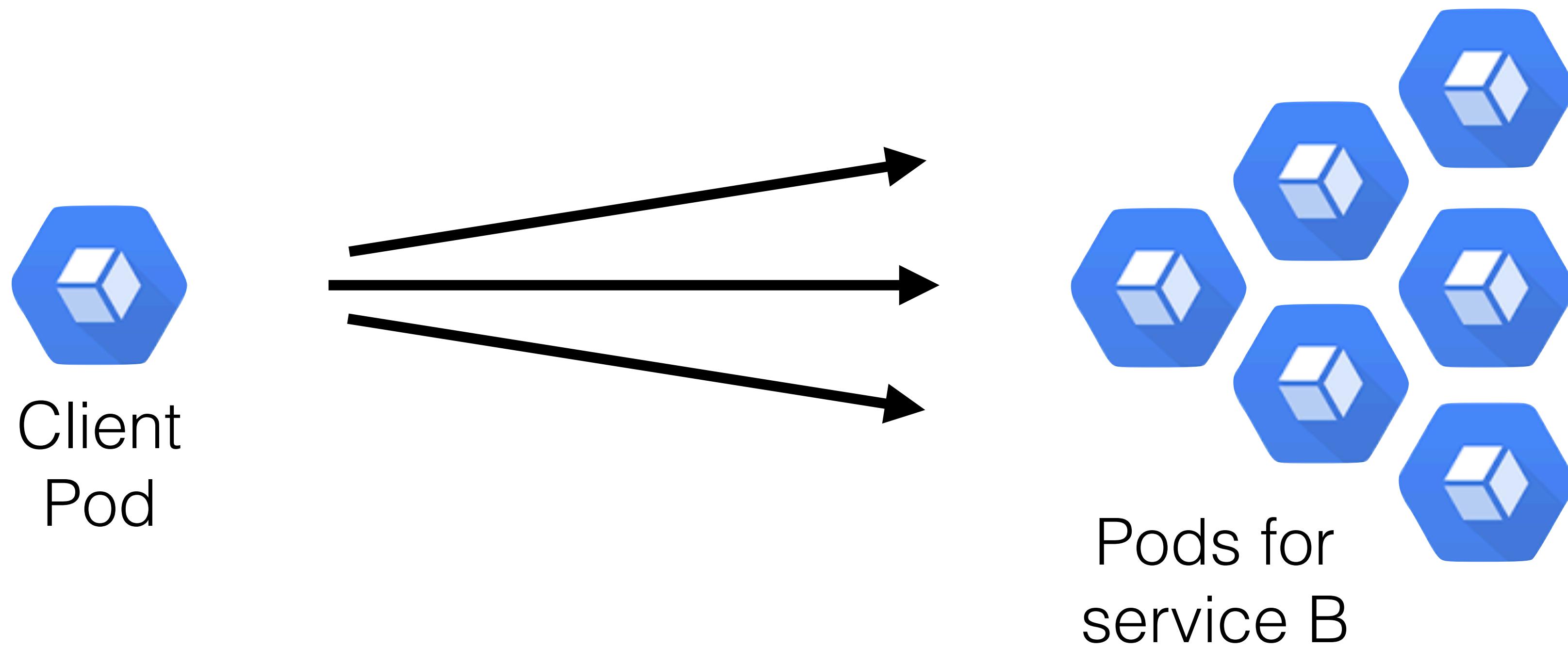
# **Scratching the surface...**





KubeResolver (I)





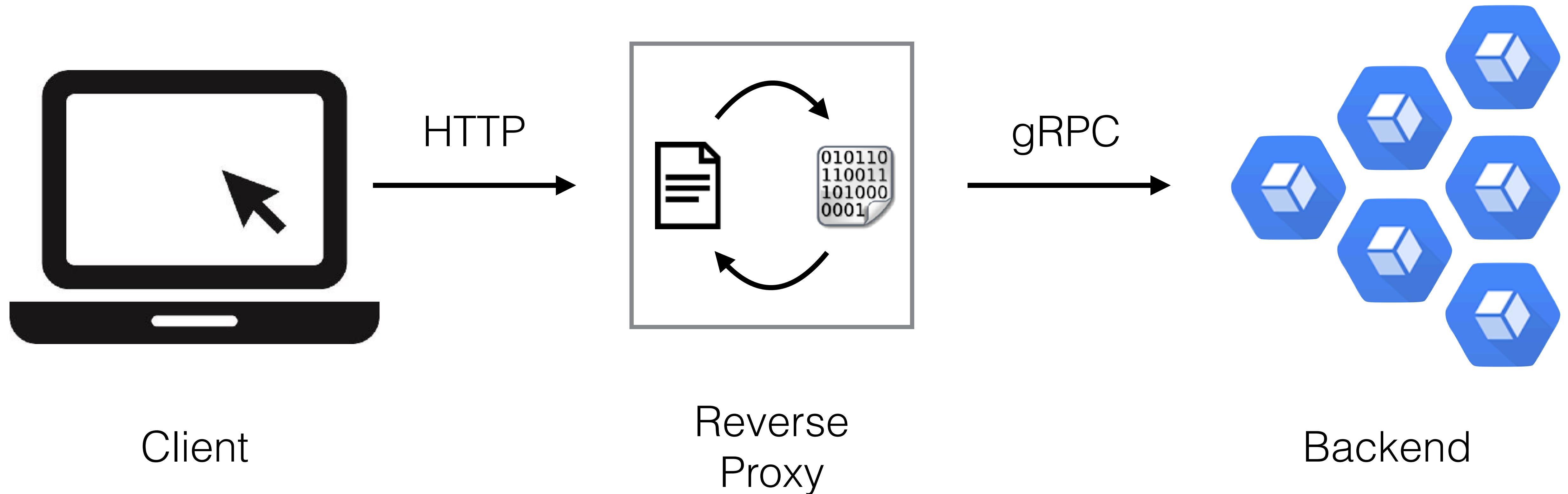
<https://github.com/sercand/kuberesolver>

KubeResolver (II)

```
balancer := kuberesolver.NewWithNamespace(namespace)
address := fmt.Sprintf("kubernetes://%s:%s", service, port)
dialOptions := []grpc.DialOption{}
conn, err := grpc.Dial(
    address, balancer.DialOption(), grpc.WithInsecure())
```



KubeResolver (III)



<https://www.weave.works/turtles-way-http-grpc/>

HTTP over gRPC (I)

```
message HTTPRequest {  
    string method = 1;  
    string url = 2;  
    repeated Header headers = 3;  
    bytes body = 4;  
}
```

```
message HTTPResponse {  
    int32 Code = 1;  
    repeated Header headers = 2;  
    bytes body = 3;  
}
```



<https://github.com/weaveworks/common/tree/master/httpgrpc>

HTTP over gRPC (II)

```
grpcMiddleware := []grpc.UnaryServerInterceptor{
    middleware.ServerLoggingInterceptor,
    middleware.ServerInstrumentInterceptor(requestDuration),
    otgrpc.OpenTracingServerInterceptor(opentracing.GlobalTracer()),
}

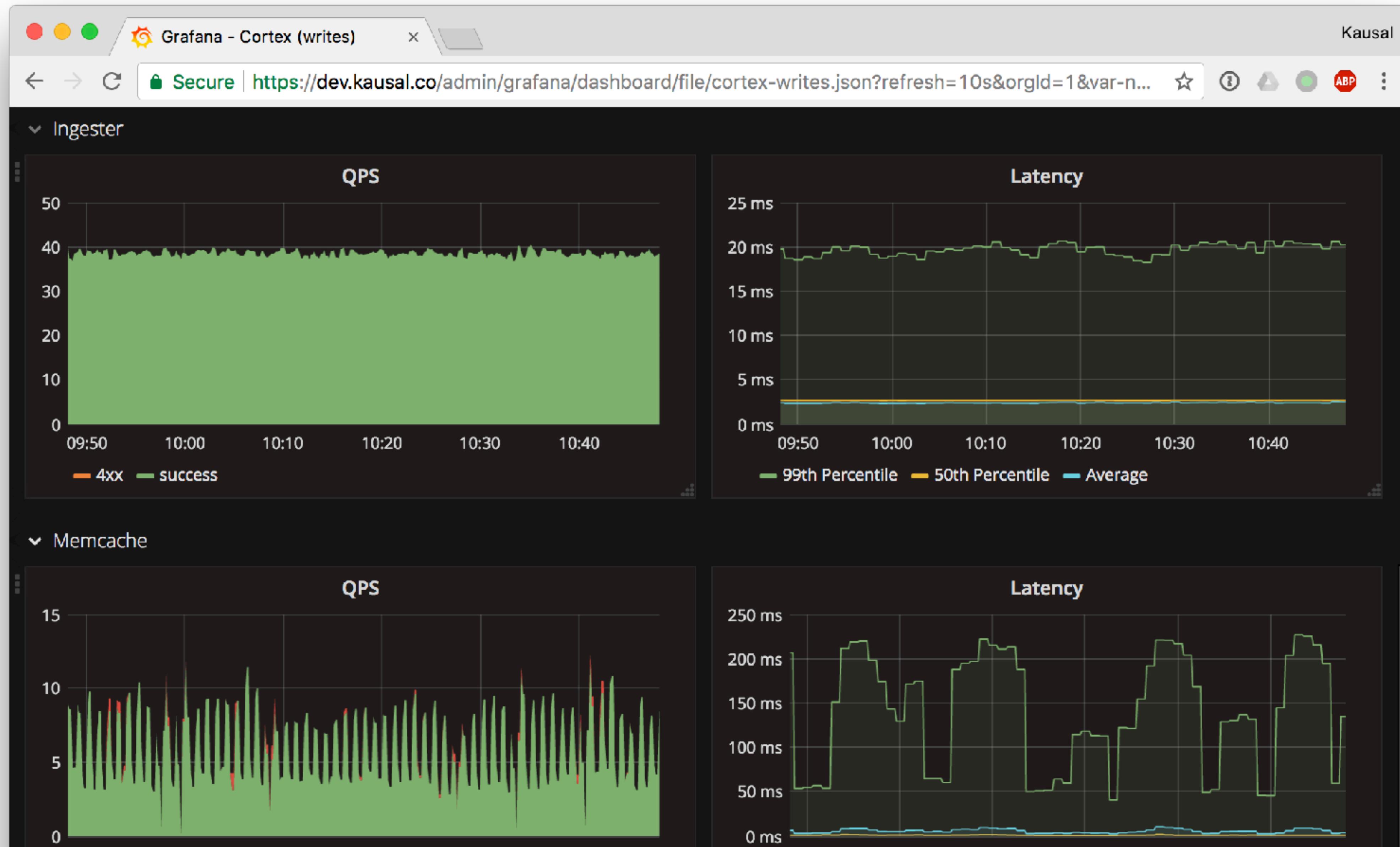
grpcServer := grpc.NewServer(
    grpc.UnaryInterceptor(grpc_middleware.ChainUnaryServer(
        grpcMiddleware...,
    )),
)
```



[https://github.com/weaveworks/common/blob/master/middleware/grpc\\_instrumentation.go](https://github.com/weaveworks/common/blob/master/middleware/grpc_instrumentation.go)

# Middleware

# Instrumentation



# **Advanced Topics**



- gRPC Streaming
- Custom Error Propagation
- Distributed Tracing
- gogoproto
- Authentication
- Timeouts
- Istio



Thanks!  
Questions?

