gRPC

Wael Kdouh - @waelkdouh

Senior Customer Engineer

v1.0

Module 2 - gRPC In Browser Apps

Module Overview

Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

Lesson: Introduction

Use gRPC In Browser Apps

- In this session we will learn how to configure an existing ASP.NET Core gRPC service to be callable from browser apps, using the gRPC-Web protocol
- gRPC-Web allows browser JavaScript and Blazor apps to call gRPC services
- It's not possible to call an HTTP/2 gRPC service from a browser-based app
- gRPC services hosted in ASP.NET Core can be configured to support gRPC-Web alongside HTTP/2 gRPC

gRPC-Web in ASP.NET Core vs. Envoy

- There are two choices for how to add gRPC-Web to an ASP.NET Core app:
    - Support gRPC-Web alongside gRPC HTTP/2 in ASP.NET Core. This option uses middleware provided by the Grpc.AspNetCore.Web package

    - Use the Envoy proxy's gRPC-Web support to translate gRPC-Web to gRPC HTTP/2. The translated call is then forwarded onto the ASP.NET Core app

- There are pros and cons to each approach. If an app's environment is already using Envoy as a proxy, it might make sense to also use Envoy to provide gRPC-Web support

- For a basic solution for gRPC-Web that only requires ASP.NET Core, Grpc.AspNetCore.Web is a good choice

Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

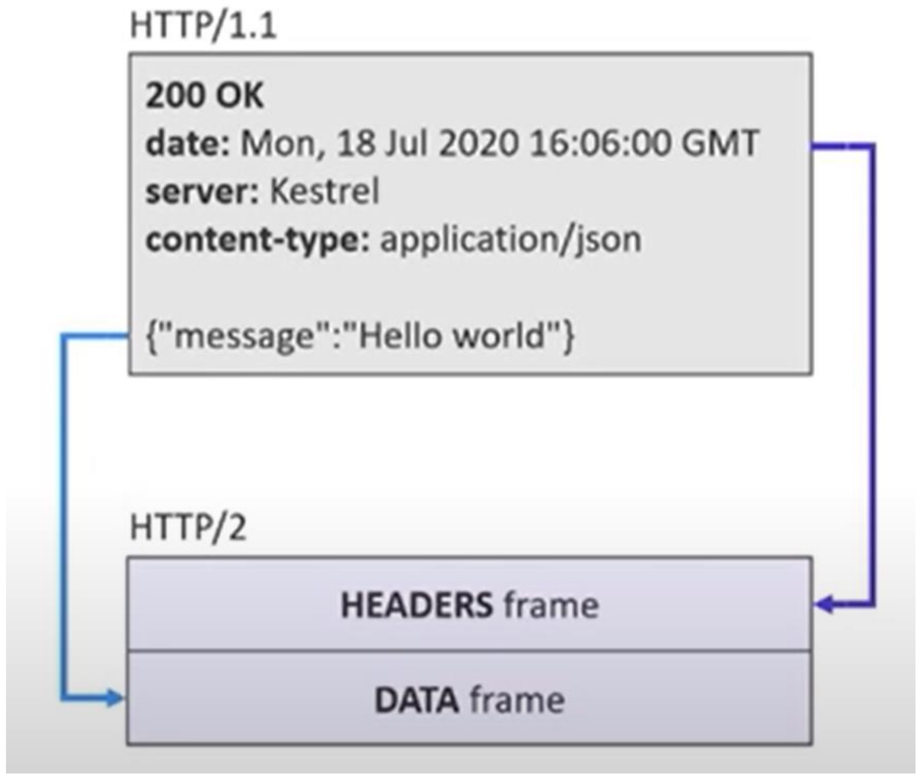
Lesson: Using gRPC With Web Apps

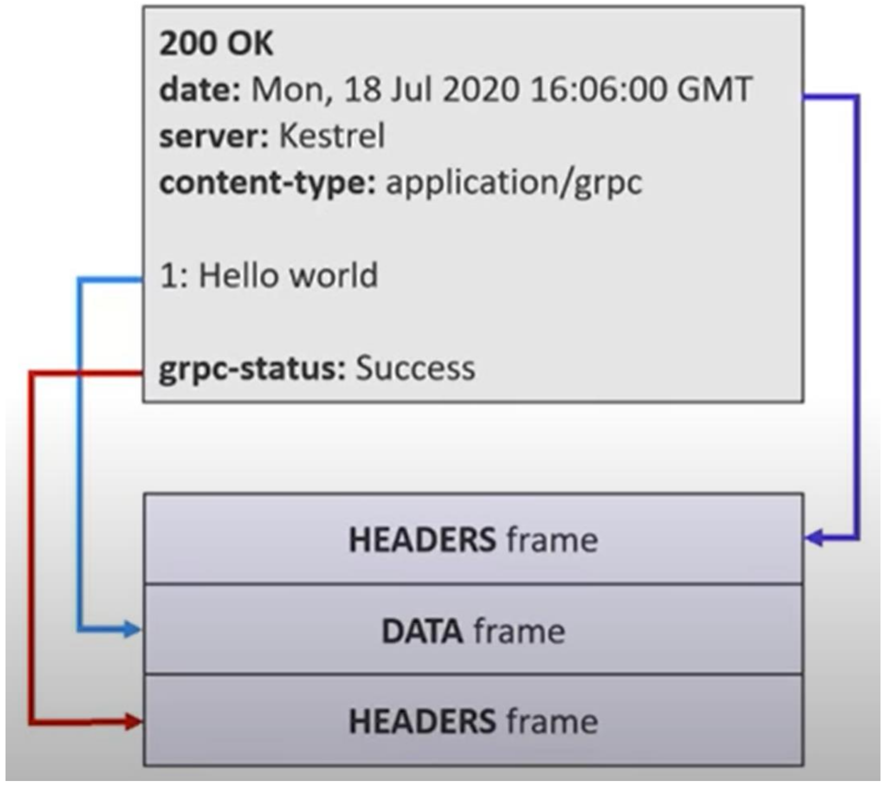
Using gRPC With Web Apps

- It is currently impossible to implement the HTTP/2 gRPC spec 3 in the browser
  - There is simply no browser API with enough fine-grained control over the requests
  - For example there is no way to force the use of HTTP/2, and even if there was, raw HTTP/2 frames are inaccessible in browsers

Using gRPC With Web Apps

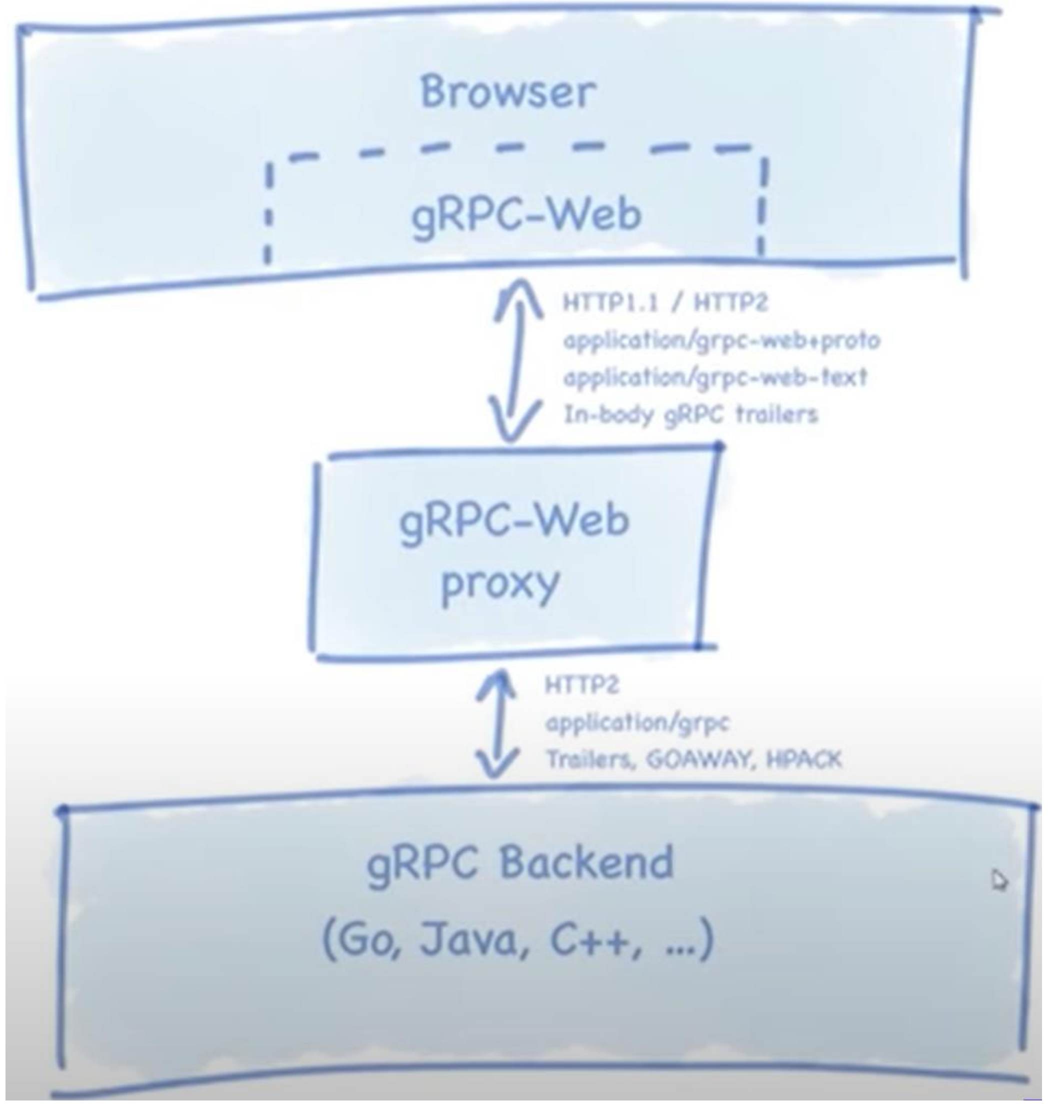
- Exploring HTTP responses


HTTP/1.1
200 OK
date: Mon, 18 Jul 2020 16:06:00 GMT
server: Kestrel
content-type: application/json
{"message":"Hello world"}
HTTP/2
HEADERS frame
DATA frame



gRPC-Web

- Transform gRPC requests
  - Move HTTP/2 trailers into response body
  - Encode messages in base64
- Envoy Proxy transforms requests going to server
- Proxy works with all gRPC implementations



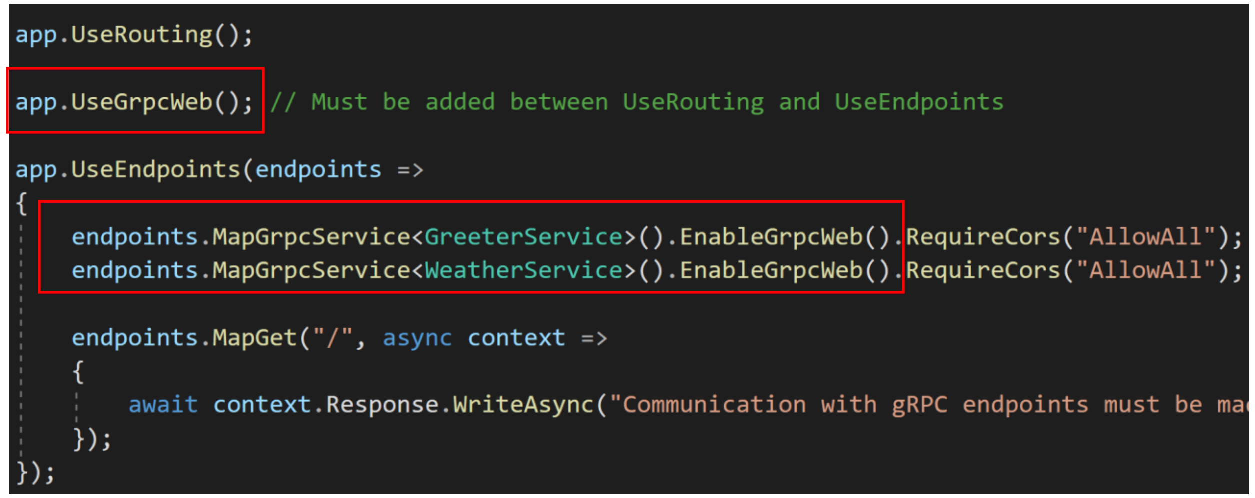
Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

Lesson: Configure gRPC-Web in ASP.NET Core

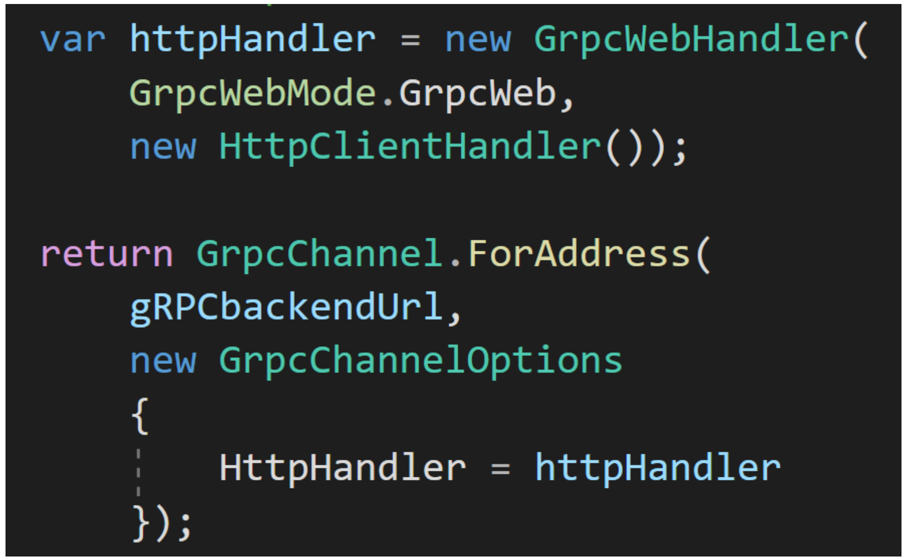
Configure gRPC-Web in ASP.NET Core

- gRPC-Web server support
  - ASP.NET middleware
  - Converts incoming gRPC-Web to gRPC
  - In-process proxy



```
app.UseRouting();

app.UseGrpcWeb(); // Must be added between UseRouting and UseEndpoints

app.UseEndpoints(endpoints =>
{
    endpoints.MapGrpcService<GreeterService>().EnableGrpcWeb().RequireCors("AllowAll");
    endpoints.MapGrpcService<WeatherService>().EnableGrpcWeb().RequireCors("AllowAll");

    endpoints.MapGet("/", async context =>
    {
        await context.Response.WriteAsync("Communication with gRPC endpoints must be ma
    });
});
```

- gRPC-Web client support
  - DelegatingHandler
  - Converts outgoing gRPC to gRPC-Web
  - Supports Blazor WebAssembly

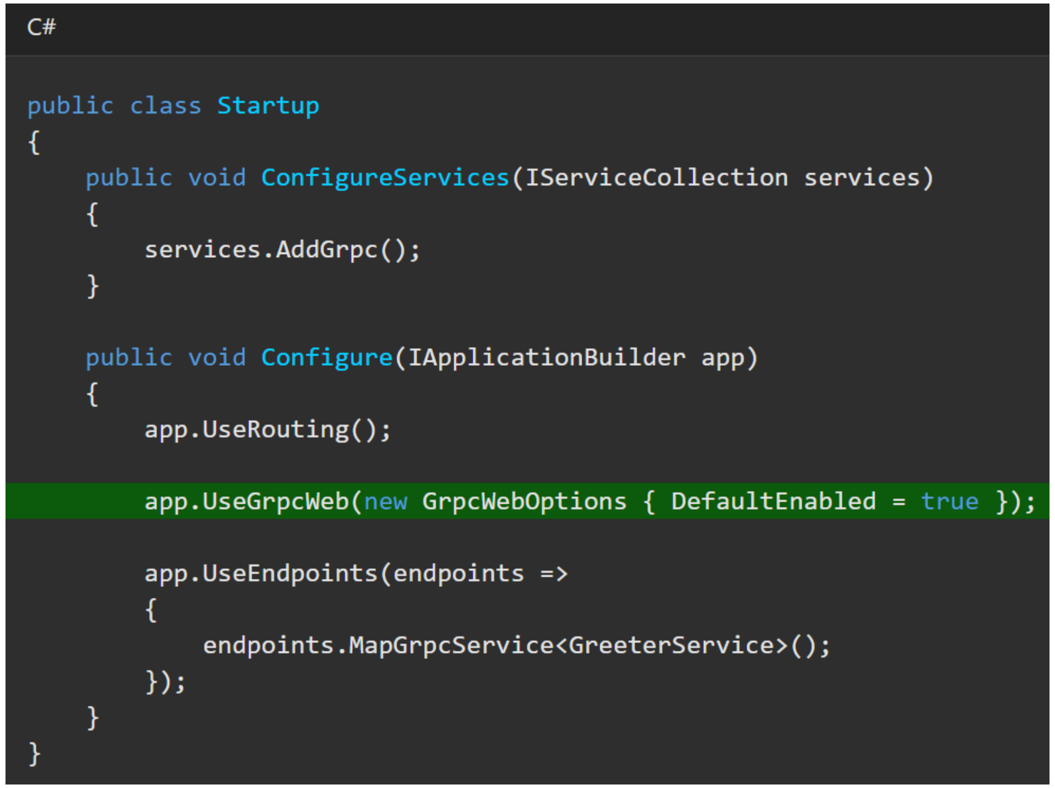


```
var httpHandler = new GrpcWebHandler(
    GrpcWebMode.GrpcWeb,
    new HttpClientHandler());

return GrpcChannel.ForAddress(
    gRPCbackendUrl,
    new GrpcChannelOptions
    {
        HttpHandler = httpHandler
    });
```

Configure gRPC-Web in ASP.NET Core

- Alternatively, the gRPC-Web middleware can be configured so all services support gRPC-Web by default and EnableGrpcWeb isn't required

- Specify new GrpcWebOptions { DefaultEnabled = true } when the middleware is added



C#

```csharp
public class Startup
{
    public void ConfigureServices(IServiceCollection services)
    {
        services.AddGrpc();
    }

    public void Configure(IApplicationBuilder app)
    {
        app.UseRouting();

        app.UseGrpcWeb(new GrpcWebOptions { DefaultEnabled = true });

        app.UseEndpoints(endpoints =>
        {
            endpoints.MapGrpcService<GreeterService>();
        });
    }
}
```

gRPC-Web

- gRPC-Web is designed to make gRPC available in more scenarios. These include:
  - **Call ASP.NET Core gRPC apps from the browser** – Browser APIs can’t call gRPC HTTP/2. gRPC-Web offers a compatible alternative.
    - JavaScript SPAs
    - .NET Blazor Web Assembly apps
  - **Host ASP.NET Core gRPC apps in IIS and Azure App Service** – Some servers, such as IIS and Azure App Service, currently can’t host gRPC services. While this is actively being worked on, gRPC-Web offers an interesting alternative that works in every environment today
  - **Call gRPC from non-.NET Core platforms** – HTTP/2 is not supported by HttpClient on all .NET platforms. gRPC-Web can be used to call gRPC services from Blazor and Xamarin

Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

Lesson: gRPC-Web and CORS

gRPC-Web and CORS

- Browser security prevents a web page from making requests to a different domain than the one that served the web page

- This restriction applies to making gRPC-Web calls with browser apps. For example, a browser app served by https://www.contoso.com is blocked from calling gRPC-Web services hosted on https://services.contoso.com

gRPC-Web and CORS

- Cross Origin Resource Sharing (CORS) can be used to relax this restriction. To allow a browser app to make cross-origin gRPC-Web calls, set up CORS in ASP.NET Core

C#

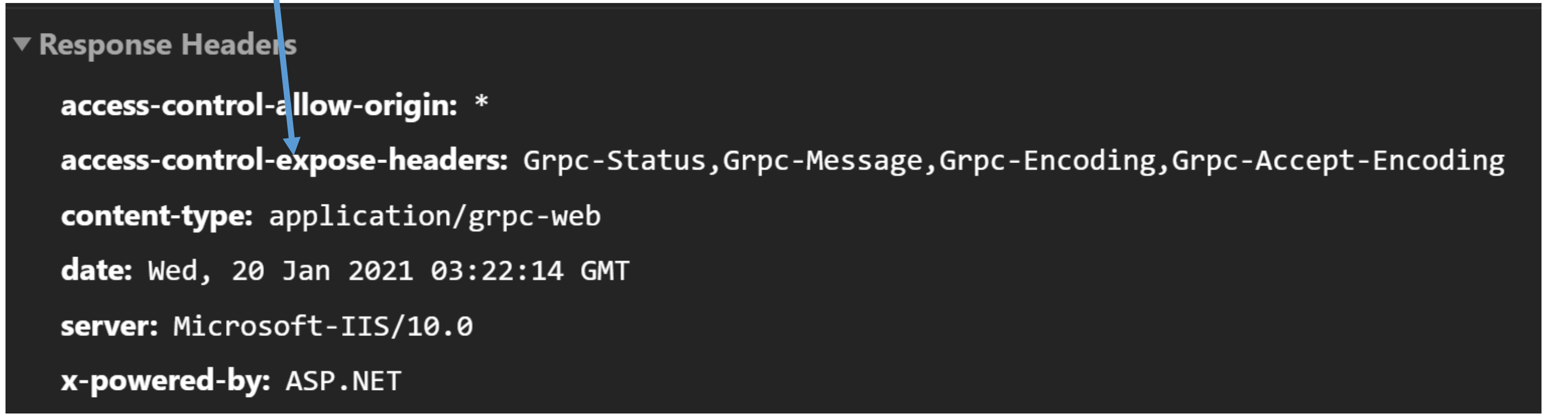
```csharp
public void ConfigureServices(IServiceCollection services)
{
    services.AddGrpc();

    services.AddCors(o => o.AddPolicy("AllowAll", builder =>
    {
        builder.AllowAnyOrigin()
               .AllowAnyMethod()
               .AllowAnyHeader()
               .WithExposedHeaders("Grpc-Status", "Grpc-Message", "Grpc-Encoding", "Grpc-Accept-Encoding");
    }));
}

public void Configure(IApplicationBuilder app)
{
    app.UseRouting();

    app.UseGrpcWeb();
    app.UseCors();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapGrpcService<GreeterService>().EnableGrpcWeb()
                                                  .RequireCors("AllowAll");
    });
}
```

What is the Purpose of WithExposedHeaders?

- It turns out that only 6 “simple” headers are allowed to be returned (for security reasons) on CORS requests. As listed in the spec, they are:
    - Cache-Control
    - Content-Language
    - Content-Type
    - Expires
    - Last-Modified
    - Pragma
- If for example you want to allow Grpc-Status", "Grpc-Message", "Grpc-Encoding", "Grpc-Accept-Encoding“ to be read, you must specify the content-length header with the Access-Control-Expose-Headers response header. The value is a comma-separated list of headers.

gRPC-Web and CORS

- .WithExposedHeaders("Grpc-Status", "Grpc-Message", "Grpc-Encoding", "Grpc-Accept-Encoding");



```
▼ Response Headers
    access-control-allow-origin: *
    access-control-expose-headers: Grpc-Status,Grpc-Message,Grpc-Encoding,Grpc-Accept-Encoding
    content-type: application/grpc-web
    date: Wed, 20 Jan 2021 03:22:14 GMT
    server: Microsoft-IIS/10.0
    x-powered-by: ASP.NET
```

Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

Lesson: gRPC-Web and Streaming

gRPC-Web and Streaming

- Traditional gRPC over HTTP/2 supports streaming in all directions. gRPC-Web offers limited support for streaming:
  - gRPC-Web browser clients don't support calling client streaming and bidirectional streaming methods
  - ASP.NET Core gRPC services hosted on Azure App Service and IIS don't support bidirectional streaming

- **<u>When using gRPC-Web, it is only recommend the use of unary methods and server streaming methods</u>**

Module 2 - gRPC In Browser Apps

Section 1: Use gRPC In Browser Apps

Lesson: Configure gRPC-Web with the .NET gRPC client

Configure gRPC-Web with the .NET gRPC client

- The .NET gRPC client can be configured to make gRPC-Web calls

- This is useful for Blazor WebAssembly apps, which are hosted in the browser and have the same HTTP limitations of JavaScript code. Calling gRPC-Web with a .NET client is the same as HTTP/2 gRPC. The only modification is how the channel is created.

- To use gRPC-Web:
  - Add a reference to the Grpc.Net.Client.Web package.
  - Ensure the reference to Grpc.Net.Client package is 2.29.0 or greater.
  - Configure the channel to use the GrpcWebHandler:

What is Blazor WebAssembly?

- Refer to the Blazor WebAssembly Deck.

Module 2 - gRPC In Browser Apps

Section 2: gRPC Vs. Rest

Lesson: Performance Comparison

Is gRPC An Alternative To Rest?

- Rest will coexist with gRPC

gRPC Remote Procedure Call

- Contract first (proto file)
- Contract is for humans
- Hides remoting complexity

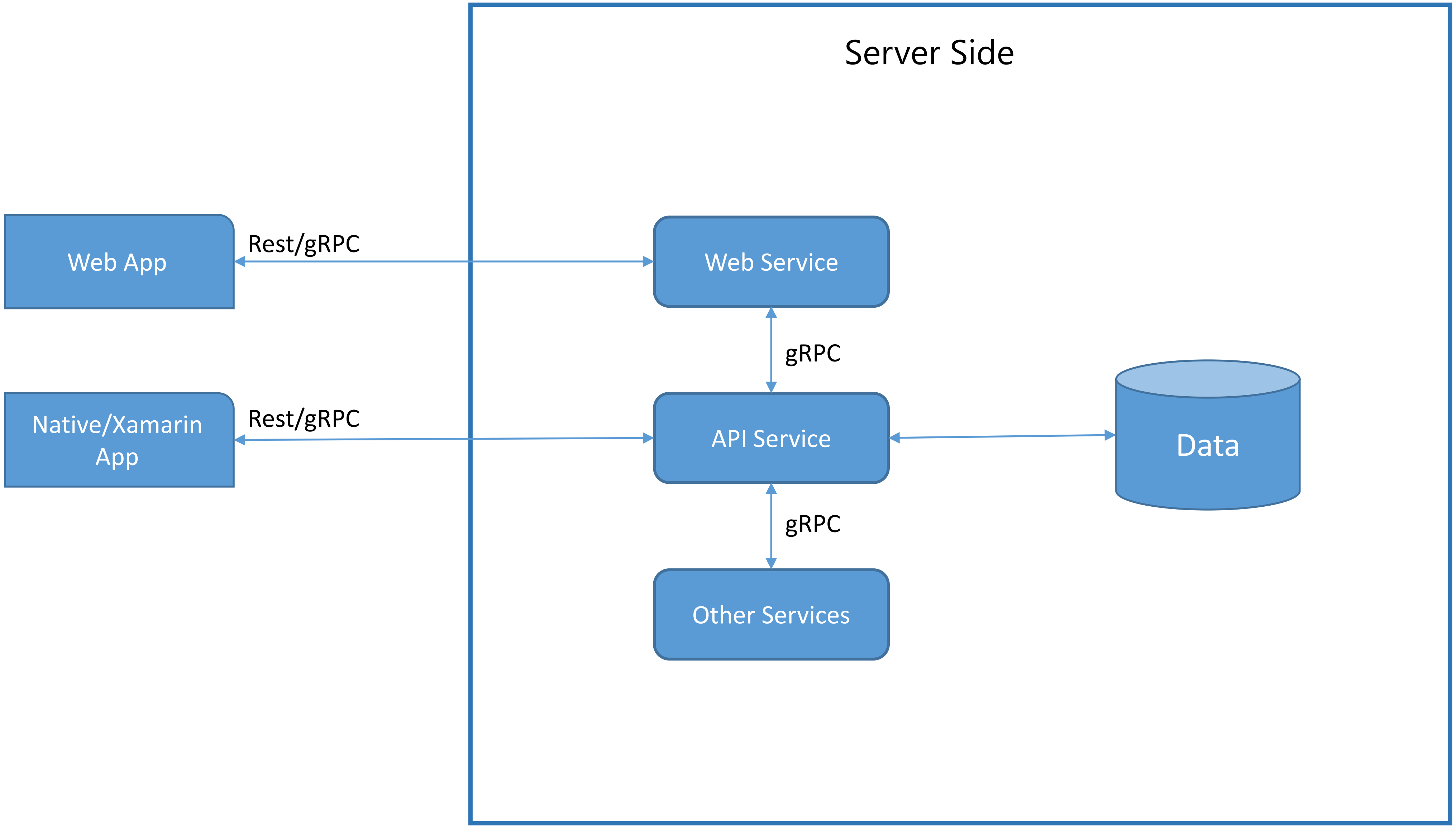
HTTP APIs

- Content first (URL, HTTP method, JSON)
- Content is for humans
- Emphasizes HTTP

- Performance
- Developer Productivity

- Widest Audience
- Ease of getting started

Is gRPC An Alternative To Rest?



Demo: gRPC Web vs Rest

Module Summary

- In this module, you learned about:
    - Using gRPC In Browser Apps
    - gRPC Vs. Rest

Lab 2: Use gRPC In Browser Apps

Microsoft