

gRPC & .NET Core 3

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t jechtom

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.dotNETportal.cz

gRPC

Open-source RPC framework (Remote Procedure Calls)

What does g stand for in gRPC?

- 1.0 'g' stands for 'gRPC'
- 1.1 'g' stands for 'good'
- 1.2 'g' stands for 'green'
- 1.3 'g' stands for 'gentle'
- 1.4 'g' stands for 'gregarious'

.....

- 1.20 'g' stands for 'godric'
- 1.21 'g' stands for 'gandalf'
- 1.22 'g' stands for 'gale'

Main gRPC features

- Contract-first
- High performance and compact serialization
- Full duplex streaming
- Tooling for many platforms

Protocol Buffers v3 (proto3)

```
syntax = "proto3";
```

```
message SearchRequest {  
    string query = 1;  
    int32 page_number = 2;  
    int32 result_per_page = 3;  
}
```

Protocol Buffers v3 (proto3)

```
...  
message SearchRequest {  
    string query = 1;  
    int32 page_number = 2;  
    int32 result_per_page = 3;  
    enum Corpus {  
        UNIVERSAL = 0;  
        WEB = 1;  
        IMAGES = 2;  
        VIDEO = 3;  
    }  
    Corpus corpus = 4;  
}
```

Protocol Buffers v3 (proto3)

...

```
message SearchResponse {  
    repeated Result results = 1;  
}  
  
message Result {  
    string url = 1;  
    string title = 2;  
    repeated string snippets = 3;  
}
```

Protocol Buffers v3 (proto3)

...

```
message SearchResponse {  
  message Result {  
    string url = 1;  
    string title = 2;  
    repeated string snippets = 3;  
  }  
  repeated Result results = 1;  
}
```


Protocol Buffers v3 (proto3)

...

```
message SampleMessage {  
  oneof result_oneof {  
    ErrorMessage error = 1;  
    SuccessMessage message = 2;  
  }  
}
```

Protocol Buffers v3 (proto3)

...

```
import "Protos/Fields.proto";
```

```
message SampleMessage {  
    map<string,Fields.Field> fields = 1;  
}
```

Protocol Buffers v3 (proto3)

```
syntax = "proto3";
```

```
Package MyApp;
```

```
message SearchRequest { ... }
```

```
message SearchResponse { ... }
```

```
service SearchService {  
    rpc DoSearch (SearchRequest)  
        returns (SearchResponse) {}  
}
```

demo

gRPC server a klient

Important built-in types

```
import "google/protobuf/empty.proto";  
import "google/protobuf/timestamp.proto";  
import "google/protobuf/duration.proto";  
import "google/protobuf/any.proto";  
import "google/protobuf/api.proto";
```

Officially supported platforms

- C++
- C#
- Dart
- Go
- Java
- Node
- Objective-C
- PHP
- Python
- Ruby
- WebJS



NodeJS klient

Why HTTP/2?

- Duplex streaming
- Multiplexing
- Compact – binary + headers compression
- Channel security with TLS
 - Required by browsers

gRPC options

- Channel credentials
 - Clients certificate
- Channel options
 - Timeouts, limits
- Call options
 - Metadata (headers)
 - Buffer hint
 - Deadline (TTL)
 - Credentials
 - ContextPropagationToken



Options and interceptors

gRPC performance

- Multiplexing within single TCP connection
- Very compact default serialization
- Horizontal scaling, HA, SSL offloading and routing possible with reverse proxy



Multiplexing

Streaming modes

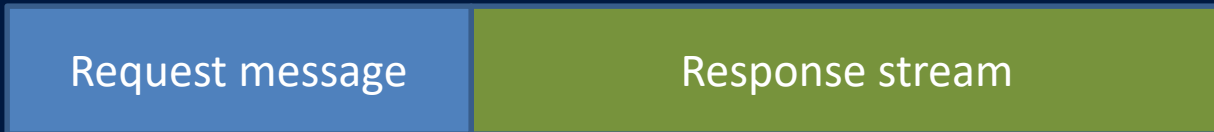
rpc Method (Req) returns (Res);



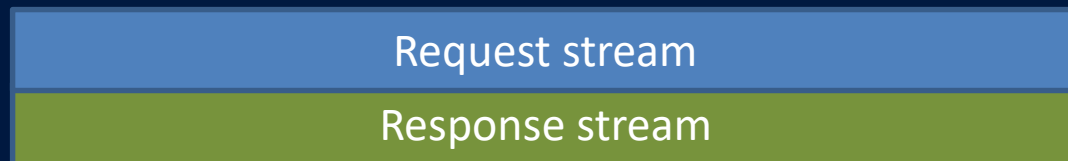
rpc Method (stream Req) returns (Res);



rpc Method (Req) returns (stream Res);



rpc Method (stream Req) returns (stream Res);





Streaming

gRPC reflection (not ready yet)

```
services.AddGrpc();  
services.AddGrpcReflection();
```

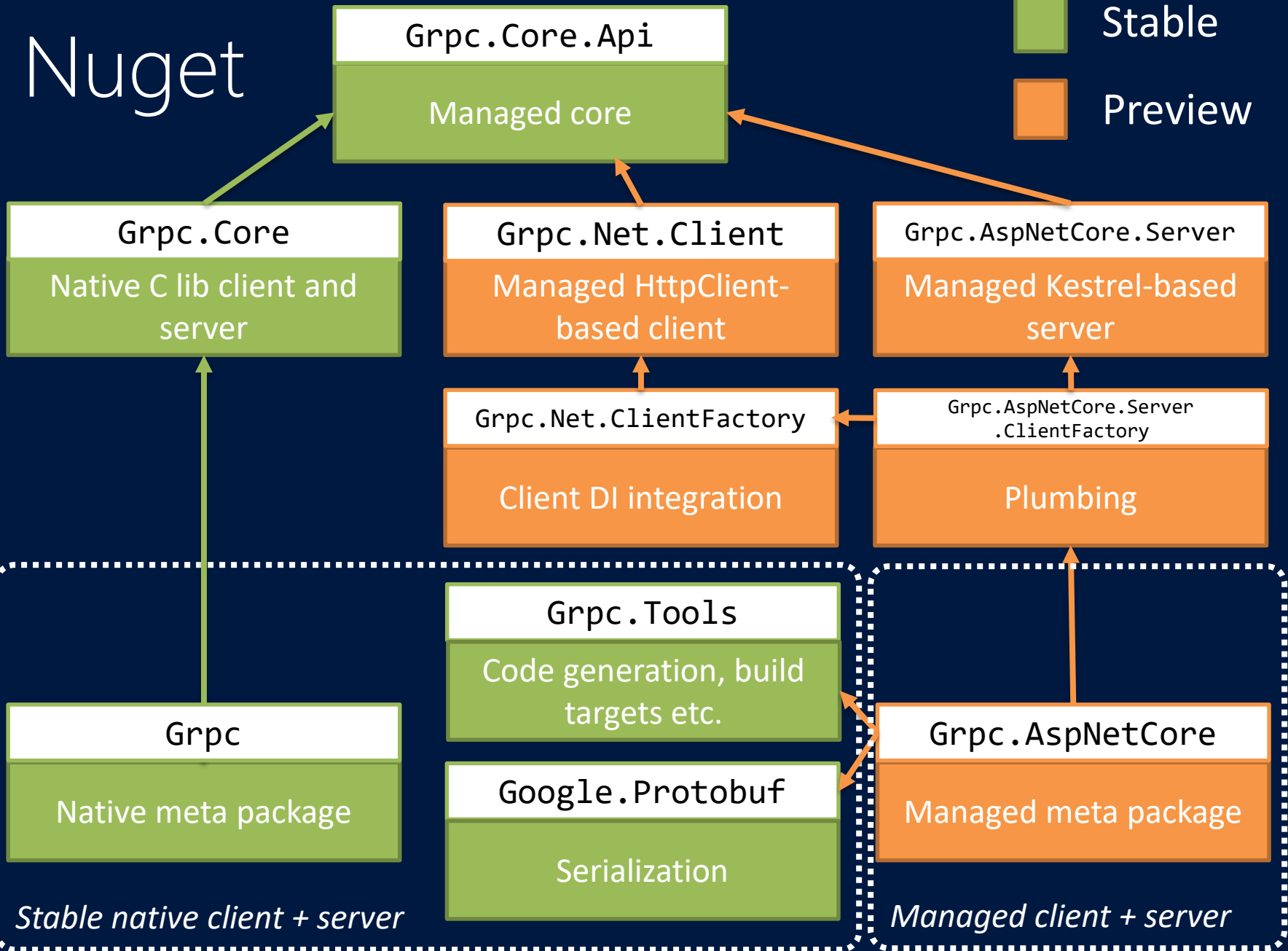
...

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

gRPC v .NET

- gRPC client + server (native C lib, stable)
 - .NET Framework 4.5+
nebo .NET Core 1.0+
 - <https://github.com/grpc/grpc/tree/master/src/csharp>
- gRPC client + server (managed, preview)
 - .NET Core 3+
 - Windows 10+ / Windows Server 2016+
nebo Linux – OpenSSL 1.0.2+
 - ASP.NET Core Kestrel server / HttpClient
 - <https://github.com/grpc/grpc-dotnet>

Nuget



When consider gRPC?

- For high performance microservices
- If regular HTTP is slow
- If strict contract is desired
- For interoperability of different platforms
- For simple message-based duplex streaming

gRPC cons?

- Not human readable, more complex
- Limited debugging tooling (for now)
- Less supported than regular HTTP+JSON
- Requires HTTP/2
- Browser support is limited

Hledám C# vývojáře do týmu v Avastu!

Tomáš Jecha (MVP)

✉ tomas@jecha.net

t [jechtom](https://twitter.com/jechtom)

e www.jecha.net



Školení vývojářů
www.dotNETcollege.cz

Články a diskuzní fóra
www.dotNETportal.cz