

Miroslav Holec

KONZULTANT PRO .NET

# Vývoj gRPC služeb & Code First

[dotnetnews.cz](http://dotnetnews.cz) & [miroslavholec.cz](http://miroslavholec.cz)

[mirek@miroslavholec.cz](mailto:mirek@miroslavholec.cz)

# Co je gRPC

Vysoce výkonný a jazykově nezávislý **open-source framework** založený na vzdáleném volání procedur (RPC).

- framework má širokou **podporu platforem**
- založen primárně na kontraktech - tzv. **contract first** (ale vč. podpory **code first**)
- komunikace je **binární** a standardizovaná skrze **Protocol Buffers**
- ze své podstaty je **bezpečný** - **vyžaduje** HTTP/2 -> **doporučuje** TLS
- **výkonný** díky podpoře HTTP/2 funkcí - streaming a binární komunikace
- snadno **škálovatelný** - jsou to HTTP služby

Autorem frameworku je Google a za dobu existence jej používá řada známých služeb.

# HTTP/2

- Release v roce 2015 – první nová verze od roku 1997 (HTTP 1/1)
- Naváže se jedno TCP spojení a skrze něj se posílají paralelně data frames v tzv. proudu (**streams**)
- **Nižší režie při vytváření TCP**, různá vylepšení (například komprese a znovupoužití headers)
- Přenos dat probíhá pomocí různých rámců (**frames**), typicky HEADERS a DATA
- Data se přenáší v **binární podobě** – menší náchylnost k chybám

## HTTP/1.x

```
POST: /upload
HTTP/1.1
Host: www.api.cz/api
Content-Type: application/json
Content-Length: 27
```

```
{ "msg" : "Hello World" }
```

## HTTP/2

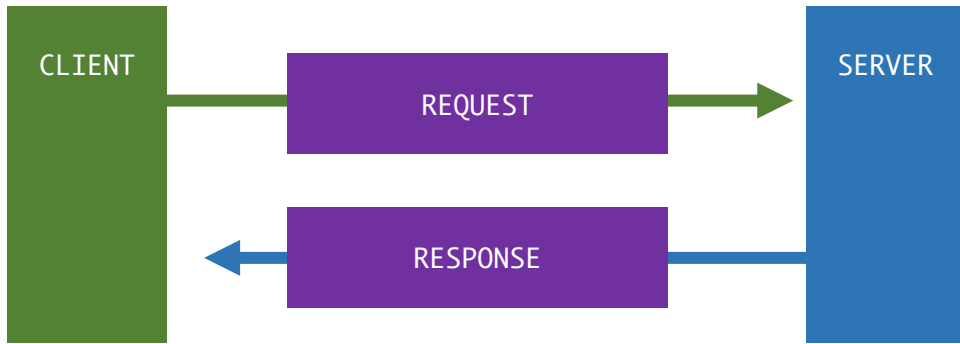
```
HEADERS FRAME
```

```
DATA FRAME
```

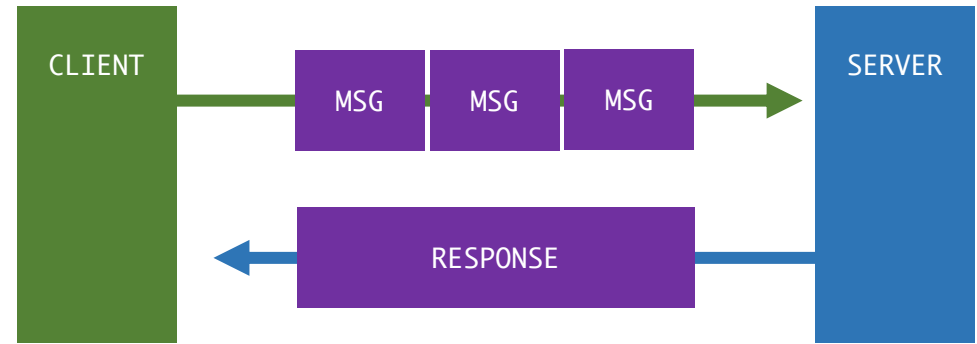
BINARY FRAMING

# Komunikace v gRPC

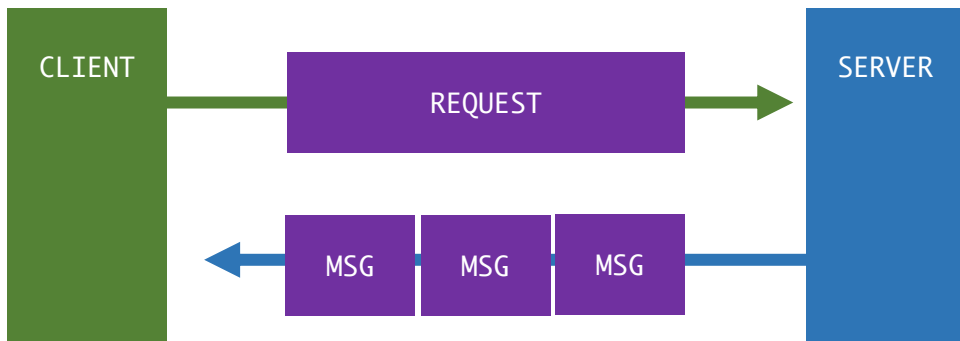
## UNÁRNÍ



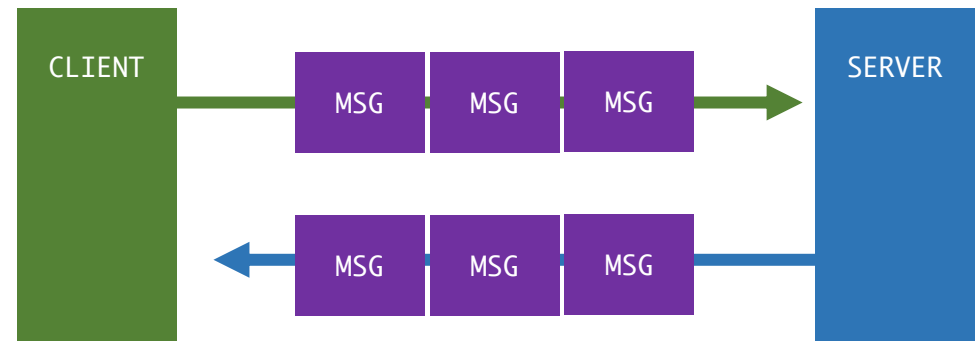
## CLIENT STREAMING



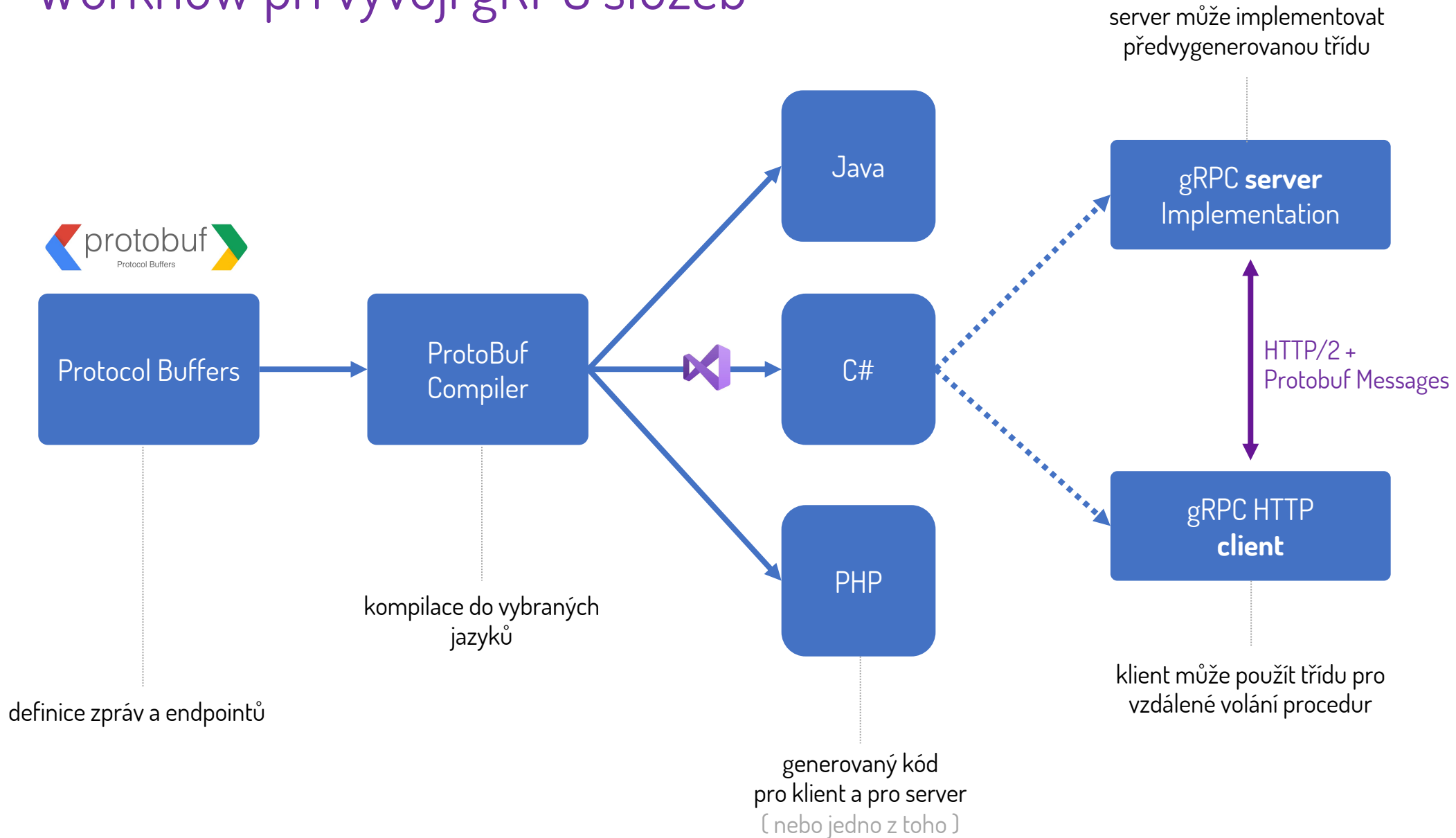
## SERVER STREAMING



## FULL DUPLEX



# Workflow při vývoji gRPC služeb



# Protocol Buffers

- tzv. interface definition language, **podklad pro generování do cílového jazyka**
- slouží k definici tzv. kontraktů, které popisují jednotlivé metody a zprávy
- **jazykově a platformně nezávislý**, zároveň rozšiřitelný
- použitelný i samostatně, například pro HTTP API
- **binární serializace** – není „lidsky“ čitelný, ale je velmi úsporný
- každá property má definované své místo v serializaci
- přípona souboru **.proto** – aktuální verze je 3



```
message Person {  
  string name = 1;  
  int32 id = 2;  
  string email = 3;  
}
```

## **pseudo serializace**

**1:** "Miroslav Holec"

**2:** 0

**3:** "mirek@miroslavholec.cz"

# Reflection

- s ohledem na vyžadovaný HTTP/2 je velmi omezené množství nástrojů
- na platformě macOS není možné použít HTTP/2 + TLS – nutné vypnout TLS
- nástroj [gRPCcli](#) zpřístupní informace o gRPC službě + podporuje export do proto
- nástroj [gRPCurl](#) dokáže konzumovat gRPC služby z příkazové řádky
- nástroj [gRPCui](#) poskytuje grafické rozhraní pro testování
- oba zmíněné nástroje potřebují definici služby

## gRPC Reflection

- řeší poskytnutí Protobuf kontraktů – automatizace v rámci konzumace
- v ASP.NET Core řešeno NuGet balíčkem [Grpc.AspNetCore.Server.Reflection](#)

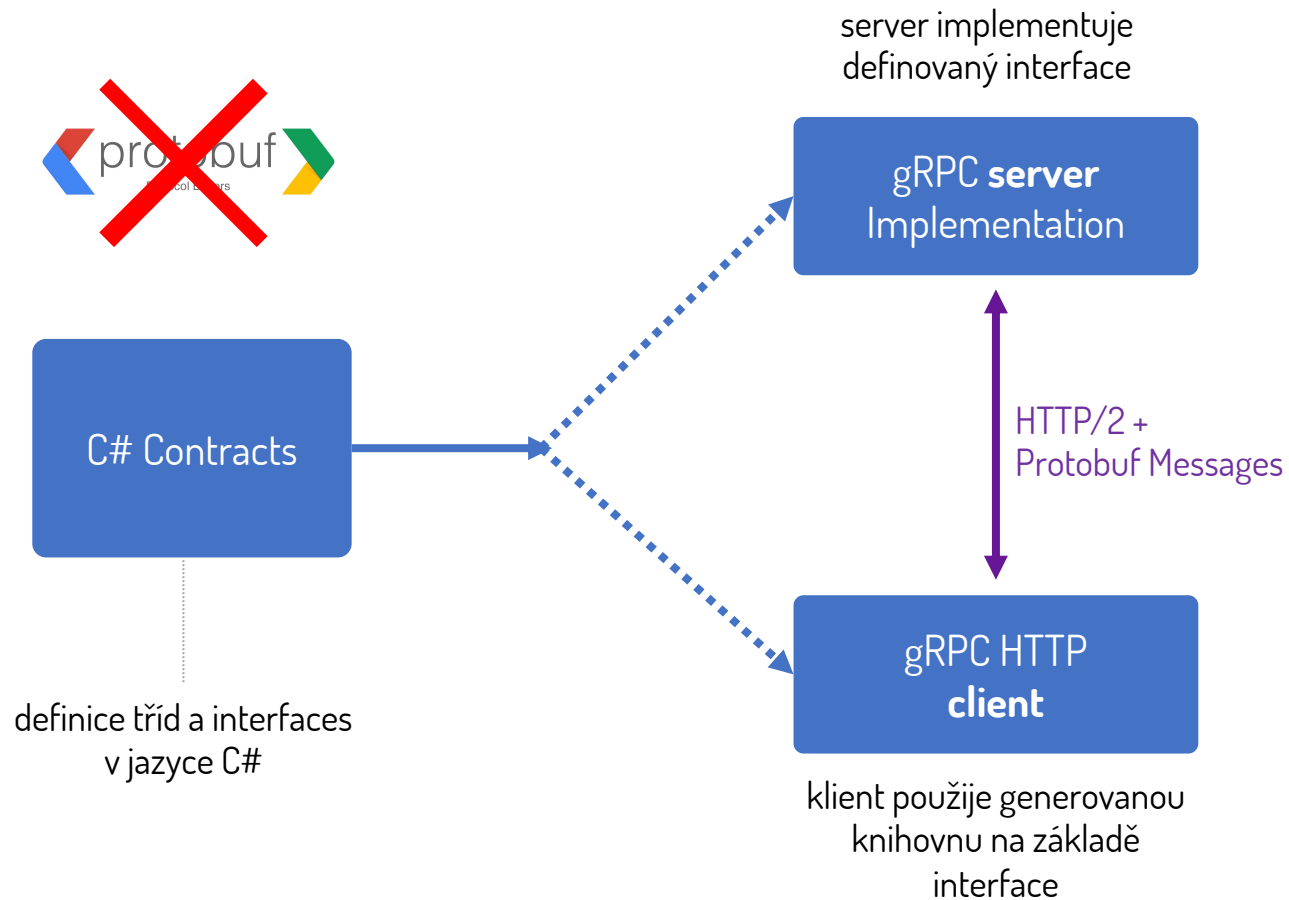
```
public void ConfigureServices(IServiceCollection services)
{
    services.AddGrpcReflection();
}
```

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

DEMO



# Vývoj gRPC Code First



- místo ProtoBuf se použije C#
- knihovna s kontrakty se sdílí (NuGet)
- klient rovnou použije DTO

Přístup je vhodný při tvorbě API v .NET ekosystému. V kombinaci s gRPC-Web lze použít též například pro aplikace Blazor WebAssembly.

DEMO

# Porovnání gRPC přístupů

Feature	gRPC Contract First	gRPC Code First
Contract	Proto soubory	Interfaces a třídy v C# Proto – lze generovat přes gRPC Reflection
Tooling	Proto compiler + NuGet pck.	NuGet pck.
Payload	Protocol buffers	Protocol buffers
Prohlížeče & JS	Bez podpory	Bez podpory
Bezpečnost	TLS	TLS
Generování klienta	Z proto souborů	Z interfaces a tříd
Podpora platform	Široké spektrum	Pouze .NET ekosystém
UI klienti pro testování	ProtoBuf / gRPC Reflection	gRPC Reflection

# Volba vhodné technologie



**REST** CRUD a webové aplikace



**SignalR** messaging, real-time komunikace



**GraphQL** efektivní dotazování nad velkým datasetem



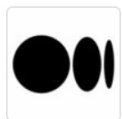
**gRPC** rychlé dotazování, výměna dat mezi systémy

# Scénáře využití gRPC služeb

- **výměna dat** mezi aplikacemi v rámci firemní infrastruktury – **code first**
- prostředí, kde oceníme **přístup contract first** - různé týmy a různé platformy
- výkonné mikroslužby vyžadující rychlou **synchronní komunikaci**
- **real-time služby**, monitorování a kontinuální přenos dat
- komunikace s **IoT zařízeními** a mobilními aplikacemi
- **blazor webassembly** s využitím gRPC-Web



Slack



medium.com



Microsoft



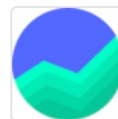
HENNGE K.K.



9GAG



Wongnai



Groww



Buzzvil



Evoqq



# Na závěr: případovka Spotify

- **200M aktivních uživatelů** měsíčně, ve špičce až **8M req/sec**
- cca **1000 vývojářů** obsluhuje téměř 3000 services
- přechod z vlastního řešení na gRPC, velkou motivací **protobuf**
- zmigrováno cca 100 gRPC služeb, cílem je bezmála 1000



## Výzvy

- akceptace nového formátu a práce s Protobuf
- sdílené repository pro všechny Protobuf soubory
- rozvoj Protobuf a zachování zpětné kompatibility
- verzování v krajní nouzi, např.: spotify.playlist.v1

Více např.: <https://www.jfokus.se/jfokus19-pres0/Adopting-gRPC-at-Spotify.pdf>



**Miroslav Holec**

MICROSOFT MVP & CONSULTANT

[mirek@miroslavholec.cz](mailto:mirek@miroslavholec.cz)

[miroslavholec.cz](http://miroslavholec.cz)

---

Dotazy