# Wysokowydajny, uniwersalny framework RPC typu open source

## gRPC w C++ z użyciem OTel

**Autorzy:**
Patryk Czuchnowski, Michał Pędrak, Andrzej Wacławik, Ivan Zarzhitski

**Rok, Grupa:**
2025, grupa 9:45 czwartek

# Spis treści

# 1 Wprowadzenie

W ramach projektu, budujemy prostą aplikację typu klient-serwer opartą o gRPC [1] z włączoną instrumentacją danych telemetrycznych (śladów, metryk, logów prowadzonej komunikacji) przy wykorzystaniu OpenTelemetry [2]. Dzięki wykorzystaniu OTLP (czyli protokołu OpenTelemetry), zebrane dane będą mogły być eksportowane do narzędzi wizualizacyjnych takich jak Grafana [3].

Celem projektu jest stworzenie aplikacji, która nie tylko umożliwia komunikację pomiędzy klientem a serwerem za pomocą gRPC, ale również zapewnia pełną obserwowalność działania systemu, czyli zdolność do zrozumienia, co dzieje się wewnątrz niego, na podstawie zewnętrznych sygnałów (danych telemetrycznych). Dzięki integracji z OpenTelemetry, aplikacja będzie monitorowana pod kątem wydajności oraz dostępności, co pozwoli na szybsze wykrywanie ewentualnych problemów.

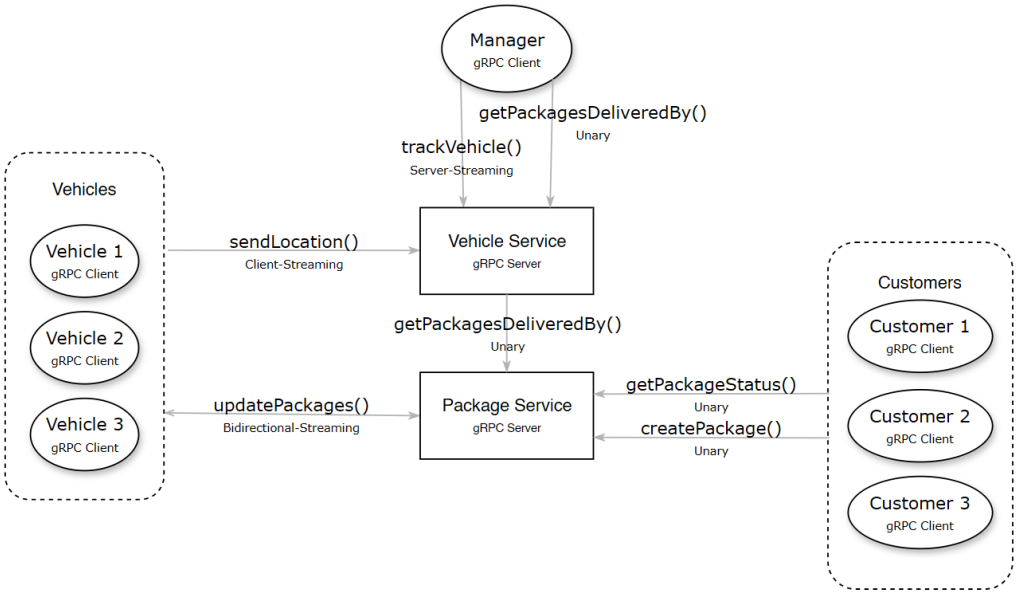
# 2 Podstawy teoretyczne i stos technologiczny

Framework gRPC to nowoczesny i wydajny system typu open source do zdalnego wywoływania procedur, który może działać w niemalże dowolnym środowisku. Umożliwia efektywne łączenie usług zarówno wewnątrz, jak i pomiędzy centrami danych, oferując możliwość podłączania modułów do obsługi równoważenia obciążenia, śledzenia danych telemetrycznych czy uwierzytelniania.

OpenTelemetry to standard i zestaw narzędzi typu open source, służący do zbierania, przetwarzania i eksportowania danych telemetrycznych z aplikacji, takich jak metryki, logi, ślady i profile. Jest on rozwijany przez Cloud Native Computing Foundation (CNCF) i ma na celu ujednolicenie sposobu obserwowalności systemów rozproszonych.

OTLP (OpenTelemetry Protocol) to standaryzowany protokół komunikacyjny używany przez OpenTelemetry do przesyłania danych telemetrycznych między aplikacjami do obserwowalności (takimi jak Grafana, Jaeger, Prometheus). Jest on binarnym protokołem opartym na gRPC lub HTTP/Protobuf [4].

W ramach projektu, wykorzystujemy technologię gRPC do komunikacji, OpenTelemetry do zbierania danych telemetrycznych. Protokół OTLP jest używany do eksportowania zebranych danych telemetrycznych do narzędzia Grafana, służącego do wizualizacji wyników. Projekt zostanie skonteneryzowany za pomocą Kubernetes oraz zdeployowany na AWS.




Rysunek 1: Diagram architektury rozwiązania

# 3 Opis koncepcji

W ramach projektu stworzymy system zarządzania flotą pojazdów i przesyłkami w architekturze rozproszonej. Kluczową rolę odgrywa w nim komunikacja między niezależnymi usługami oraz monitorowanie stanu systemu i jego komponentów w czasie rzeczywistym.

# 4 Architektura rozwiązania

Architektura systemu została przedstawiona na rys. 1 System składa się z następujących komponentów:

- **Vehicle Service** (gRPC serwer)
  Odpowiada za zarządzanie lokalizacją pojazdów.
  - **sendLocation()** – Client-Streaming
    Wywoływana przez pojazdy, pojazd wysyła strumień danych z lokalizacją do serwera.
  - **trackVehicle()** – Server-Streaming
    Menedżer otrzymuje ciągły strumień lokalizacji wskazanego pojazdu.
  - **getPackagesDeliveredBy()** – Unary
    Zwraca ile paczek zostało dostarczonych przez podany pojazd w danym dniu.

- **Package Service** (gRPC serwer)
  Zarządza paczkami i udostępnia informacje o nich klientom.
  - **updatePackages()** – Bidirectional-Streaming
    Dwukierunkowa komunikacja z pojazdem, pojazd wysyła aktualizacje statusu przesyłek, serwer odpowiada jaką paczkę dostarczyć następnie i gdzie.
  - **createPackage()** – Unary
    Wywoływana przez klienta, tworzy nową paczkę do dostarczenia – z adresem nadawcy, odbiorcy itp.
  - **getPackageStatus()** – Unary
    Wywoływana przez klienta, zwraca aktualny status paczki.
- **Pojazdy** (gRPC klienci)
  Wysyłają dane lokalizacyjne i aktualizują informacje o dostawach.
- **Menedżer** (gRPC klient)
  Może śledzić lokalizację wybranego pojazdu oraz w celu monitorowania wydajności wysyła zapytania do serwisu ile dany pojazd dostarczył paczek w danym dniu.
- **Klienci** (gRPC klienci)
  Tworzą i śledzą przesyłki.

Dzięki OpenTelemetry, serwery gromadzą dane telemetryczne, które są następnie przesyłane przez OTLP w celu eksportu do narzędzia Grafana, która umożliwia wizualizację i monitorowanie aplikacji w czasie rzeczywistym.

## Odniesienia

[1] gRPC website. https://www.cncf.io/projects/grpc/.

[2] OpenTelemetry website. https://www.cncf.io/projects/opentelemetry/.

[3] Grafana tools website. https://grafana.com/.

[4] Protobuf website. https://protobuf.dev/.