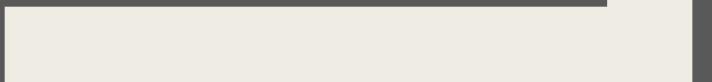


РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

Павел Кюркчиев
Ас. към ПУ „Паисий Хилендарски“
@rkyurkchiev

GRPC



Какво е Remote procedure call (RPC)?

- RPC е модел за мрежово програмиране, както и техника за комуникация между процеси, която се използва за комуникация от точка до точка между софтуерни приложения.
- RPC е протокол, посредством който една програма може да поиска услуга от друга програма, намираща се на отдалечен компютър, без да е необходимо да разбира подробностите на свързващата мрежа.

Модел на изпълнение

- RPC използва client-server модела.
- Начинът, по който работи RPC е следният: подател или клиент създава заявка под формата на процедура, функция или повикване към отдалечен сървър. Сървърът приема заявката и започва нейната обработка. Когато отдалеченият сървър обработи заявката, изпраща отговор към клиента и той продължава своя процес на работа.

- Използването на "lightweight processes" или "threads", споделящи едно и също адресно пространство, позволява няколко RPC заявки да се изпълняват едновременно.

Какво е gRPC?

- Представлява рамка за разработка на RPC системи. Първоначално рамката е изработена от Google за тяхни вътрешни нужди. Сега проектът е с отворен код.

Основни компоненти

- HTTP/2
- Protobuf serialization (proto3)
- Interface Definition Language (IDL)

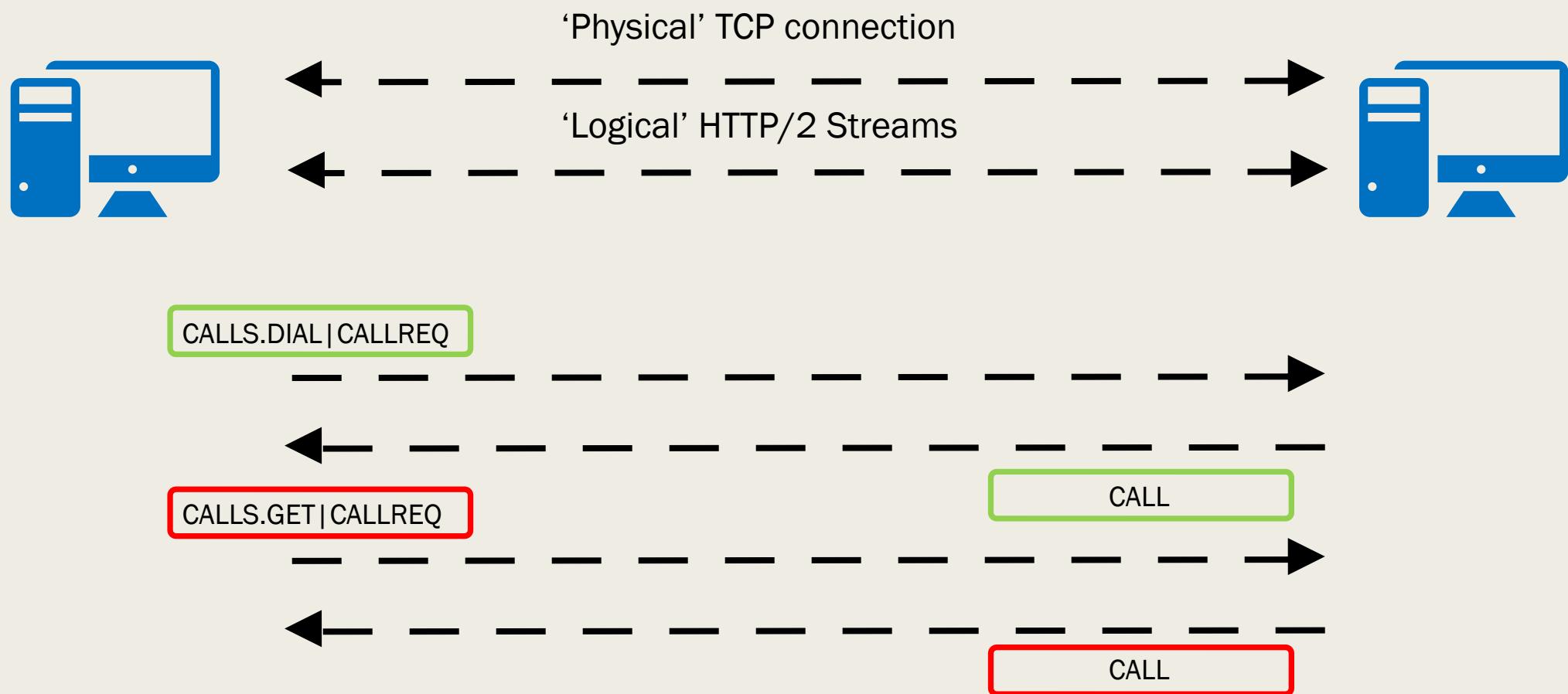
Начини на комуникация

- Unary RPC
- Server streaming RPC
- Client streaming RPC
- Bidirectional streaming RPC
 - Deadlines/Timeouts

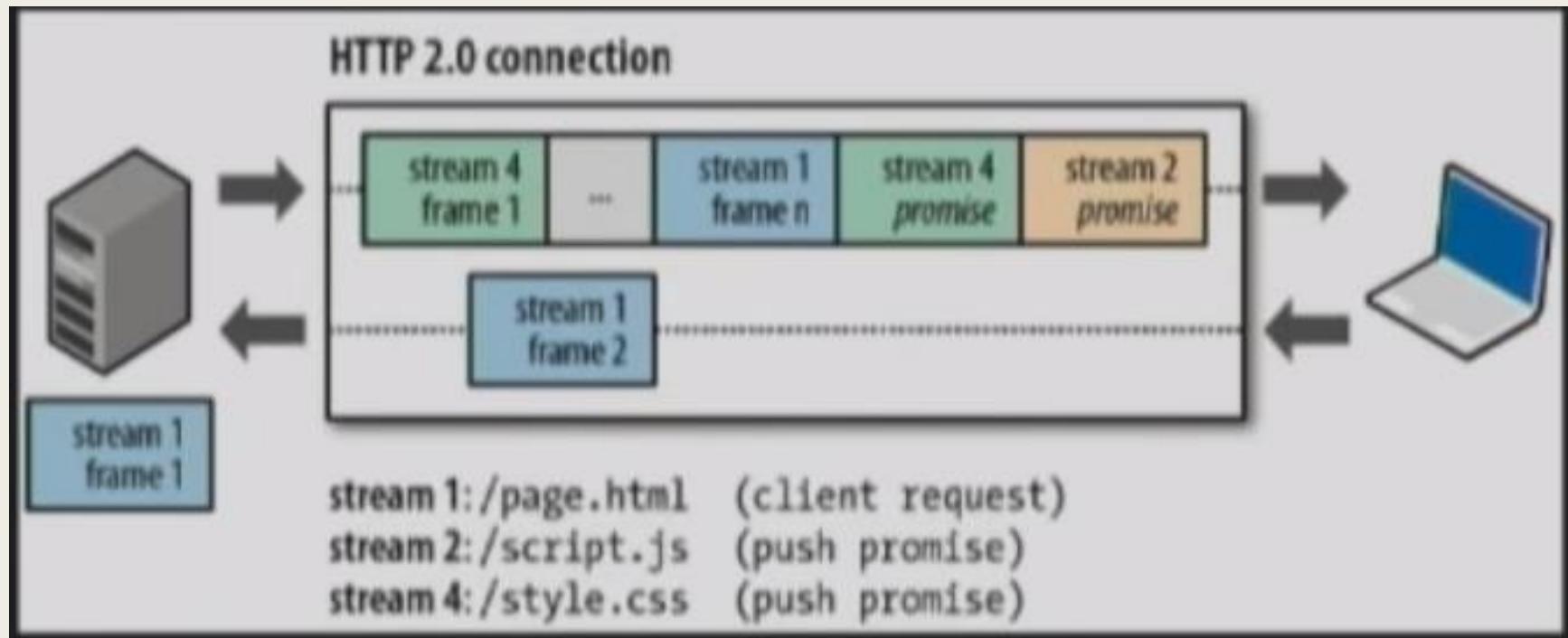
Легенда

- Unary - клиентът изпраща една заявка до сървъра и получава един отговор.
- Stream – клиентът изпраща заявка до сървъра и получава поток за четене (последователност от съобщения).

GRPC: on the wire 1



GRPC: on the wire 2



Верификация

- gRPC поддържа TLS и верификация, базирана на токени.
- TLS е криптографски протокол за предаване на информация по компютърната мрежа.

Имплементация

- Три високо производителни събитийно базирани имплементации
 - C
 - Ruby, Python, node.js, PHP, C#, Objective-C, C++ – всички те са “C” базирани.
 - Java
 - Netty + boringSSL via JNI.
 - Go
 - Чиста имплементация на Go използваща stdlib crypto/tls пакет.

RESTful vs gRPC

- Ресурсно-ориентиран
- IDL опционален
- Синхронен по подразбиране
- Unary
- Перфектен за serverless системи
- API-ориентиран
- IDL насочен
- Асинхронен по природа
- Streaming или Unary
- Приоритет е производителността

RESTful vs gRPC

- Използвай: когато трябва бързо да се създаде нещо или ако е необходимо да бъде лесно разбирамо от крайния потребител.
- Не използвай: когато се изисква проверка на типа данни или трябва да се намали размера на предаваната информация.
- Използвай: когато има комуникация между микроуслуги, намиращи се в един кълстер, или когато се изисква бързодействие.
- Не използвай: когато трябва да се предава информация между браузър и back-end услуги.

DEMO GRPC

examples/GrpcGreeter

ВЪПРОСИ ?

