

Microservices gRPC

All you need to know



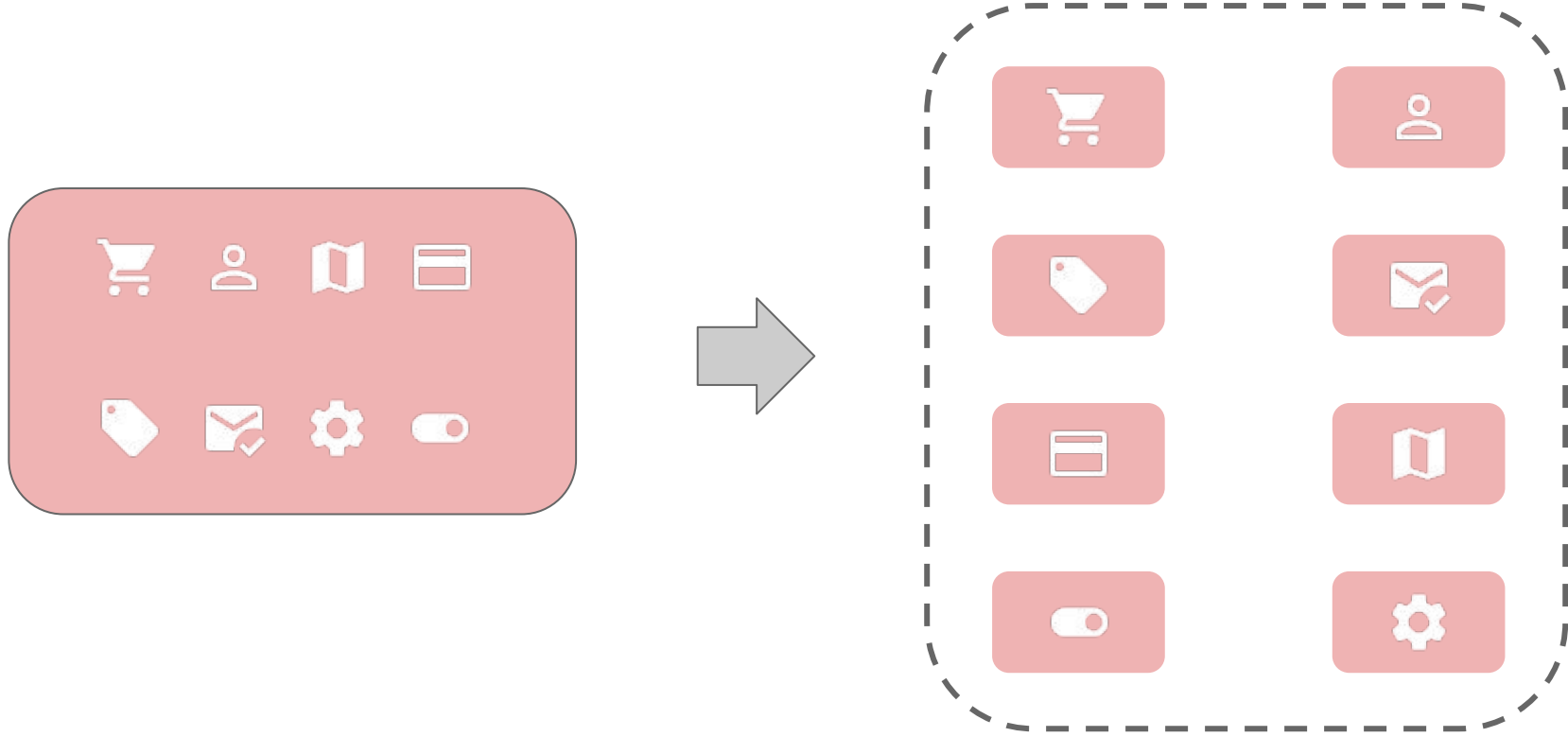
Quem é o Odair ?

- Architecture Manager na Ci&T
- Co-Host do podcast Conversa Ágil
- Quase 20 anos sendo Javeiro



A sua arquitetura de microserviços
foi escolhida corretamente ?

Monolith slicing - Once upon a time

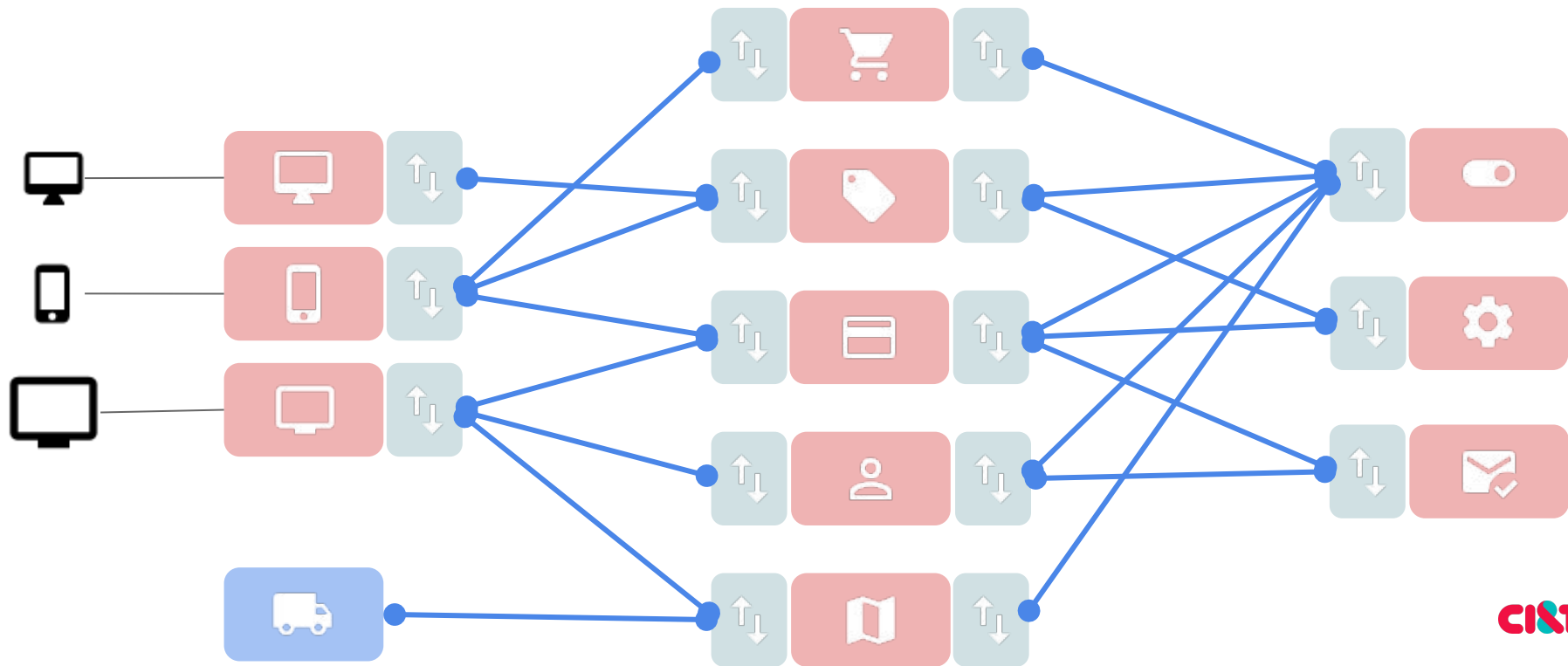


Monolith slicing - Once upon a time

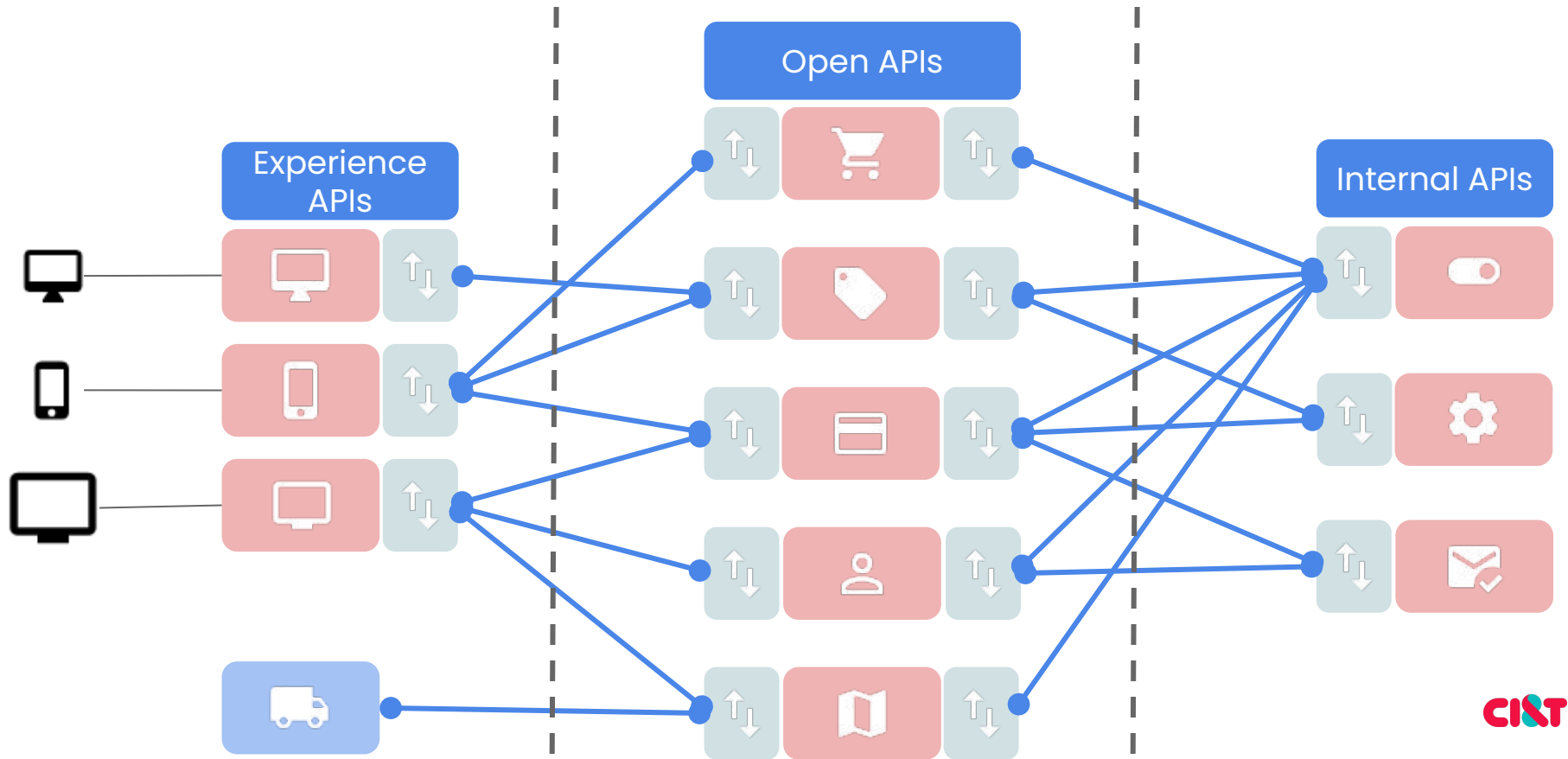


- Engenharia (linguagem, plataforma)
- Arquitetura (clean architecture)
- Protocolo de comunicação (rest)
- Tracing de log e log distribuído
- Toggles e configuração
- Lock-in cloud
- Observability (cloud offline)
- Circuit Breaker
- Service Discovery
- Database

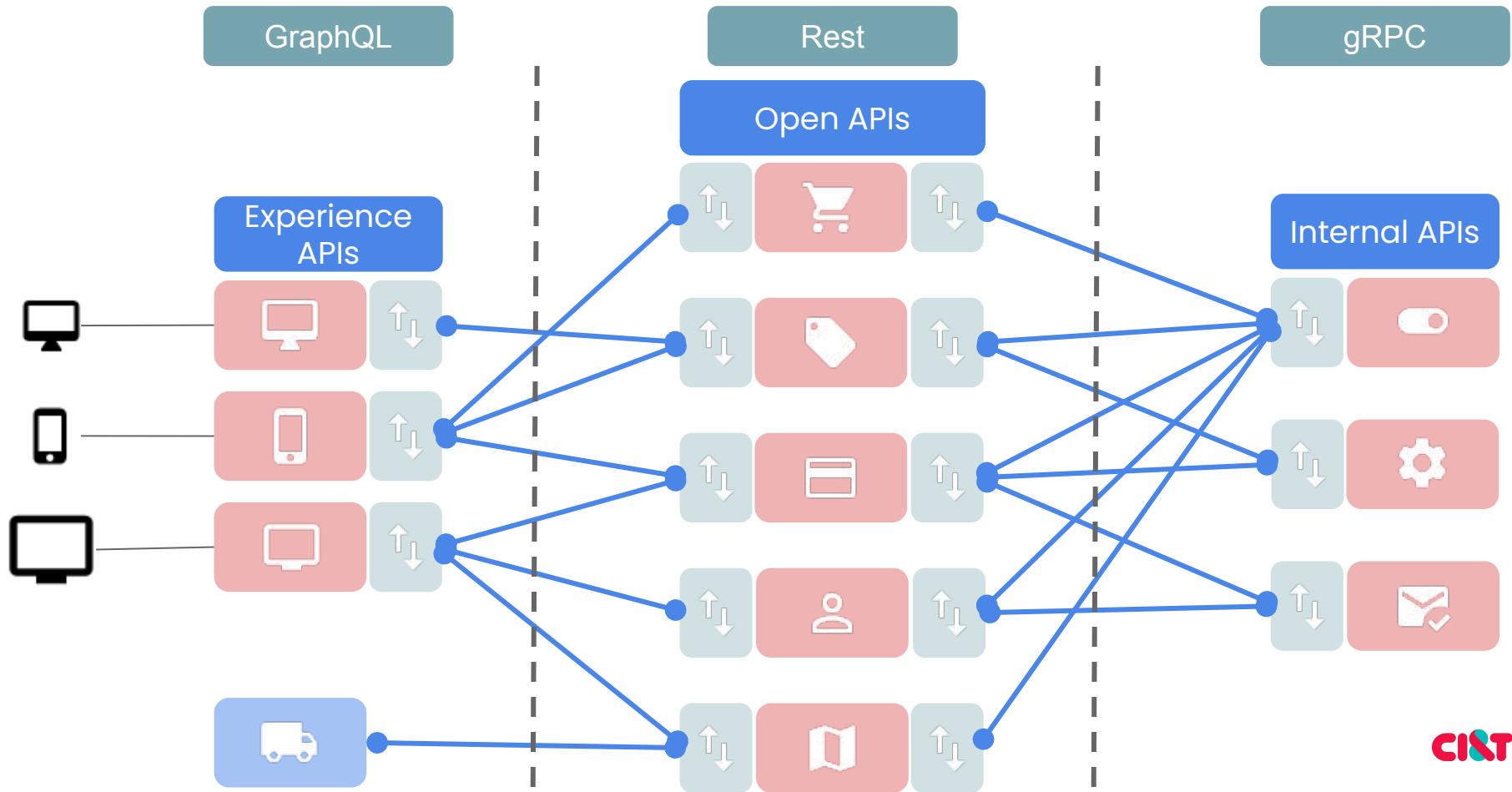
Monolith slicing



Microservices



Microservices



Software Development Architecture and Design 2019 Q1 Graph

<http://infoq.link/architecture-trends-2019>

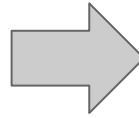
InfoQ



How it works ?

RPC: Remote Procedure Call

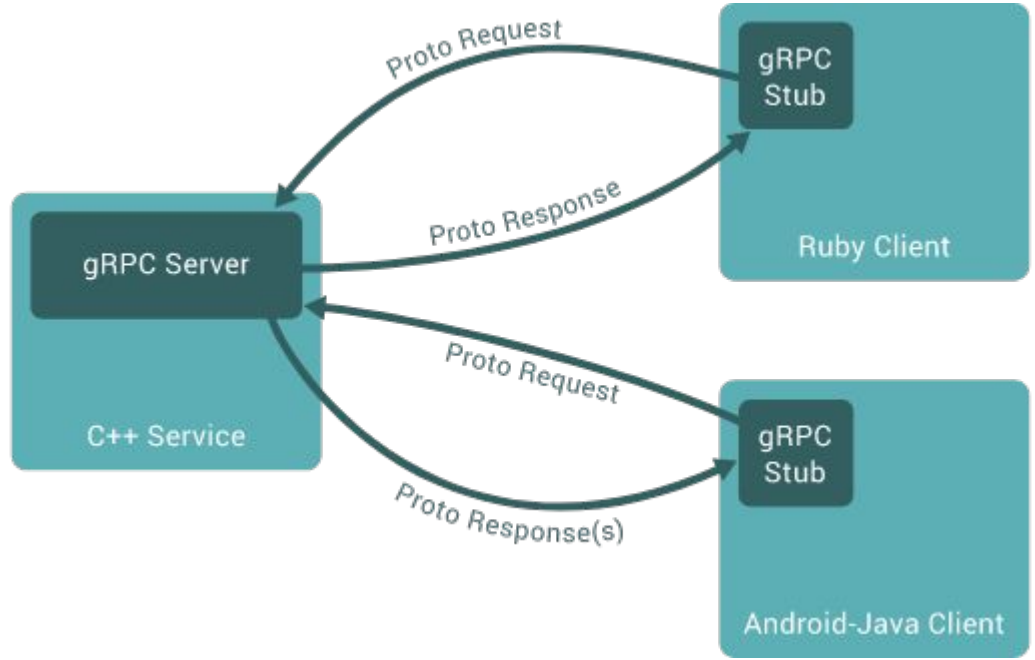
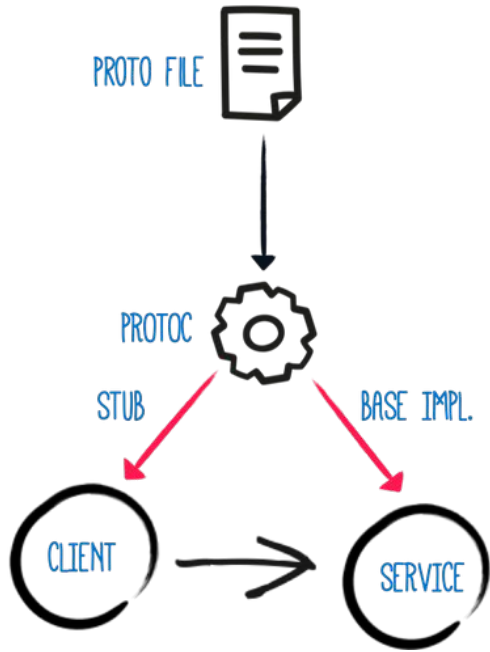
- É um protocolo request-response
- Criado em 1981
- Chamada de um "método" em um servidor (não confuda com RMI)



gRPC: Remote Procedure Call

- Google criou um framework em 2015
- Foco em performance de comunicação
- Usa apenas o protocolo HTTP/2
- Usa Protocol Buffers como linguagem IDL

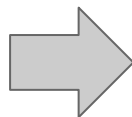
How it works ?



How it works ?

Protocol Buffers

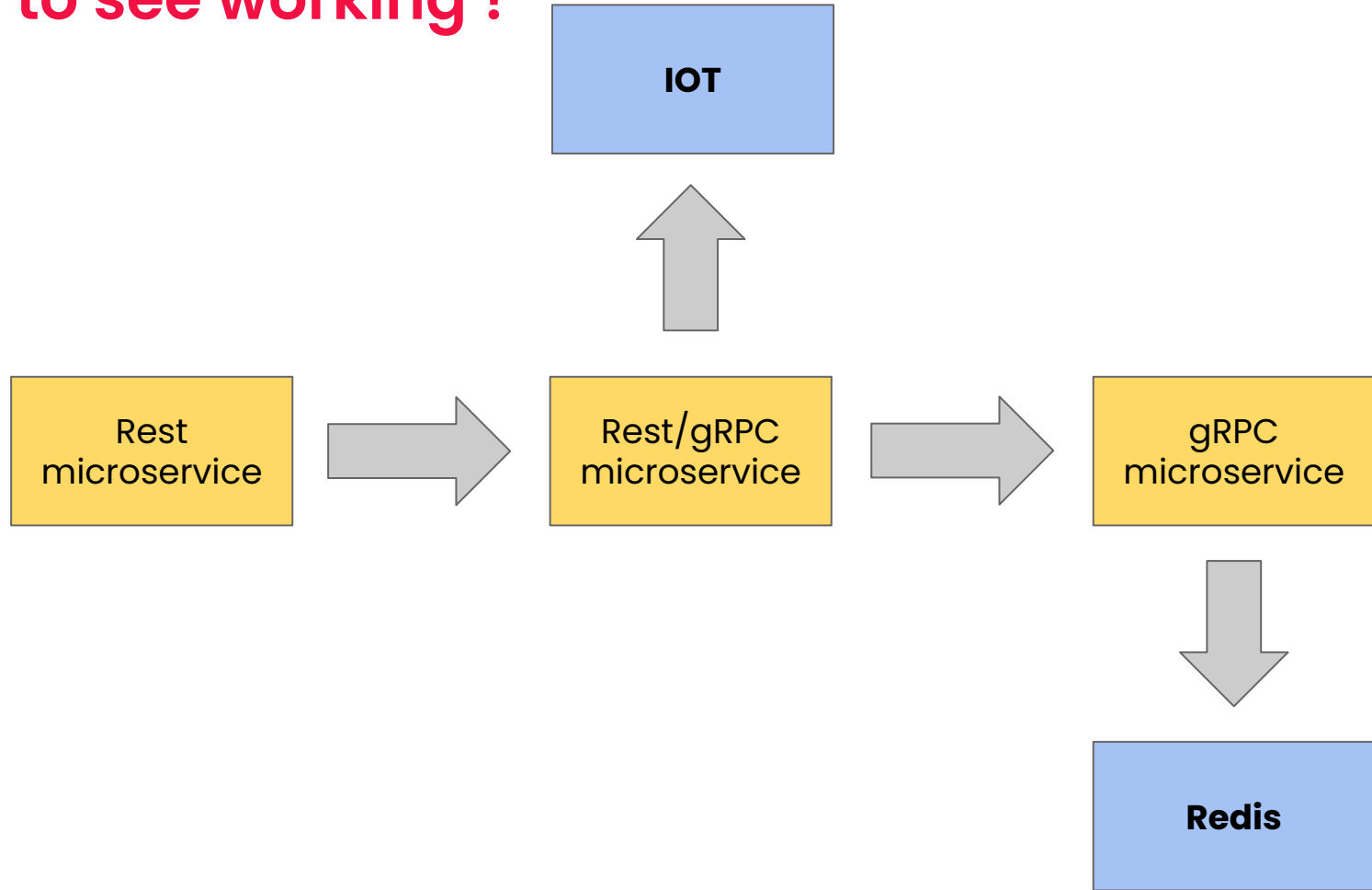
```
syntax = "proto3";  
package tutorial;  
import "google/protobuf/timestamp.proto";  
  
option java_multiple_files = true;  
option java_package = "com.tdc.java.notification.grpc";  
option java_outer_classname = "NotificationGrpcService";  
  
service NotificationService {  
    rpc sendNotification(Notification) returns (Status) {}  
}  
  
message Notification {  
    int32 id = 1; // Unique ID number for this person.  
    string from = 2;  
    string to = 3;  
    string msg = 4;  
    enum Type {  
        TELEGRAM = 0;  
        THING = 1;  
        EMAIL = 2;  
    }  
    Type notificationType = 5;  
}
```



protobuf-maven-plugin
protoc

- target/generated-sources/protobuf/java
 - com.tdc.java.notification.grpc
 - Notification.java
 - NotificationGrpcService.java
 - NotificationOrBuilder.java
 - Status.java
 - StatusOrBuilder.java

How to see working ?



How it works ?

Só acredito vendo o código ...

How it works ?

gRPC: Remote Procedure Call

- Autenticação: suporta TLS com credenciais para o canal e para a chamada
- Reduz o custo de encode/decode de mensagem (latencia)
- Arquivo proto pode gerar código para +10 linguagens
 - C++, Java, Python, Go, ...
- Permite streaming bi-direcional
- Spring usa o Netty como server

Health Check

```
grpcurl -plaintext localhost:9091 grpc.health.v1.Health/Check
```

- Consul (key-value)
- Eureka (service registration)
- Slack API
- Load Balancing

Basics <https://api.slack.com/web#basics>

The Web API is a collection of [HTTP RPC-style methods](#), all with URLs in the form `https://slack.com/api/METHOD_FAMILY.method`.

While it's not a REST API, those familiar with REST should be at home with its foundations in HTTP.

Use HTTPS, SSL, and TLS v1.2 or above when calling all methods. [Learn more about our SSL and TLS requirements.](#)

Each method has a series of arguments informing the execution of your intentions.

Microservicos é a solução para todos os problemas?

ciandt.com

THANK YOU

 Driven by **Impact**