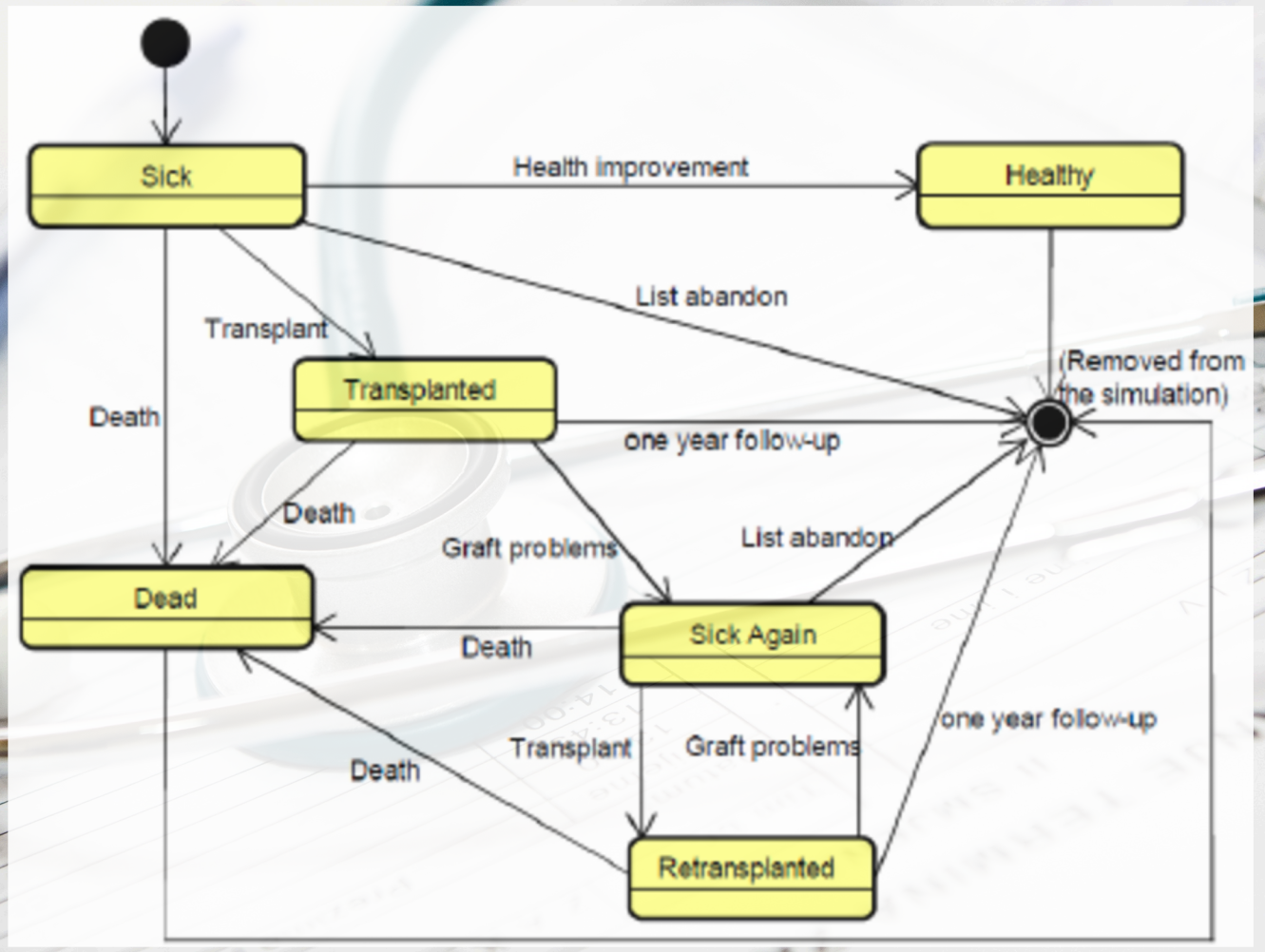
Desafios da transição de estado em um mundo serverless

Alex Coqueiro
Head de Arquitetura de Soluções para América Latina, Canada e Caribe
AWS Public Sector

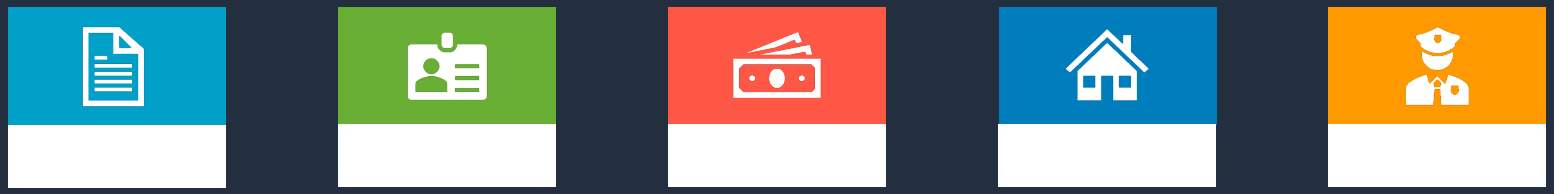
@alexbcbr

... Transição de estado em ação no dia-a-dia ...

Transição de estado adiciona contexto de negócio

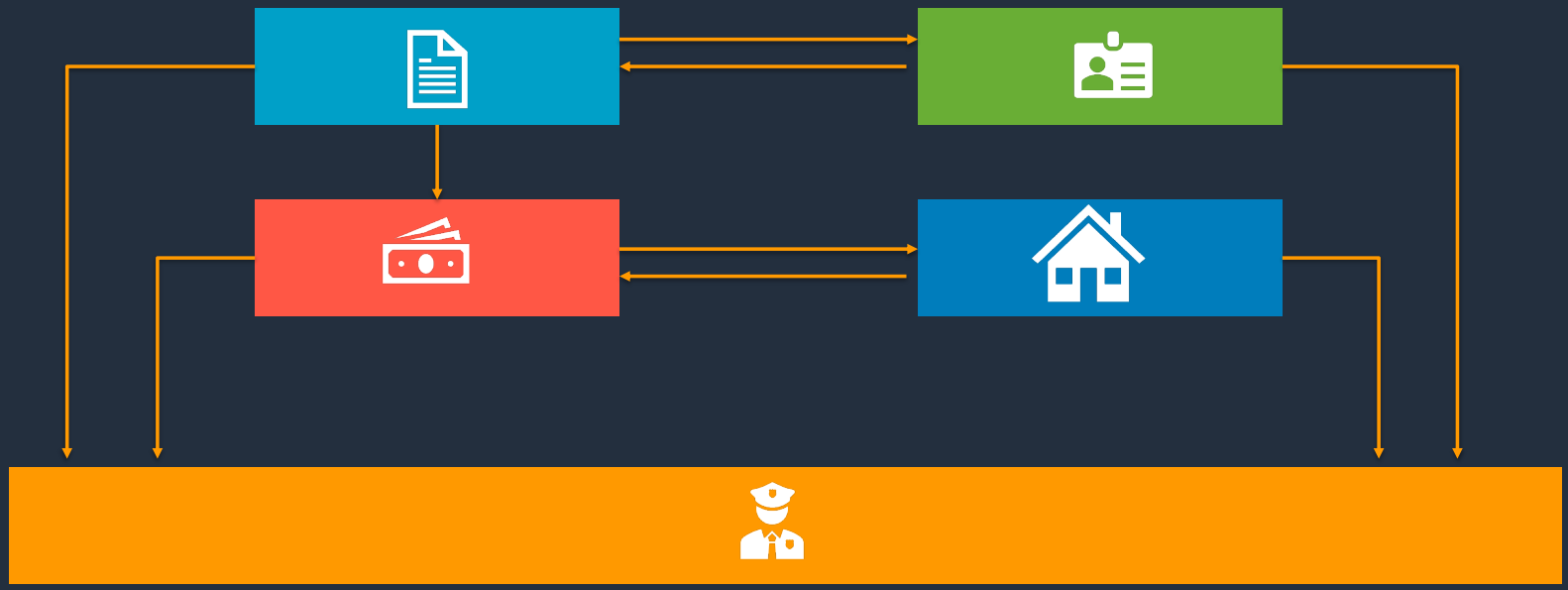


Mas como pensar em “Microserviços” & “Serverless” neste cenário



Mas como pensar em “Microserviços” & “Serverless” neste cenário

Mas agora quem é o responsável por controlar a transição de estado?



Microservices Patterns (http://microservices.io)

Microservice Architecture

Supported by Kong

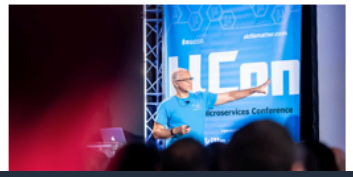
Patterns | Articles | Presentations | Resources | Assessment Platform new | Adopt new | Refactoring new | Testing new | Other Languages

About Microservices.io

Microservices.io is brought to you by Chris Richardson. Experienced software architect, author of POJOs in Action, the creator of the original CloudFoundry.com, and the author of Microservices patterns.

Chris helps clients around the world adopt the microservice architecture through consulting engagements, and training classes and workshops.

New virtual bootcamp: Distributed data patterns in a microservice architecture

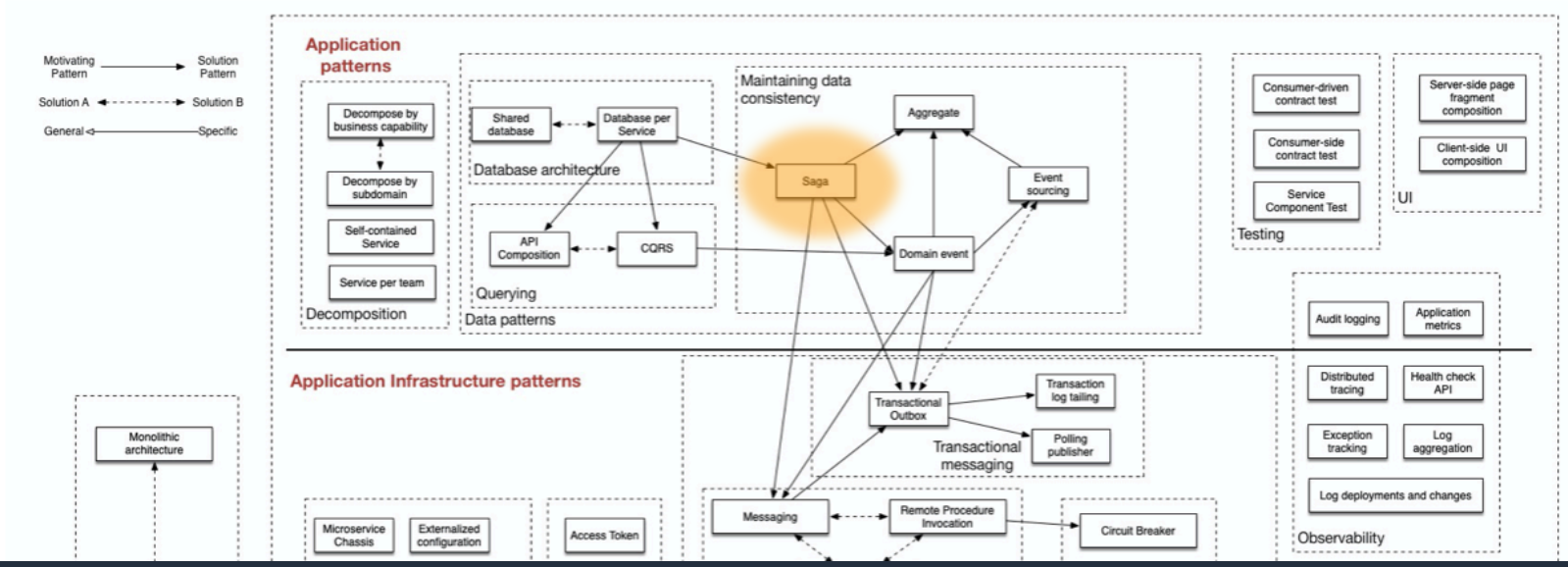


A pattern language for microservices

The beginnings of a pattern language for microservice architectures.

点击这里，访问本系列文章的中文翻译

Click here for Chinese translation of the patterns



Pattern aplicado a Transição de Estado

Saga

Event source

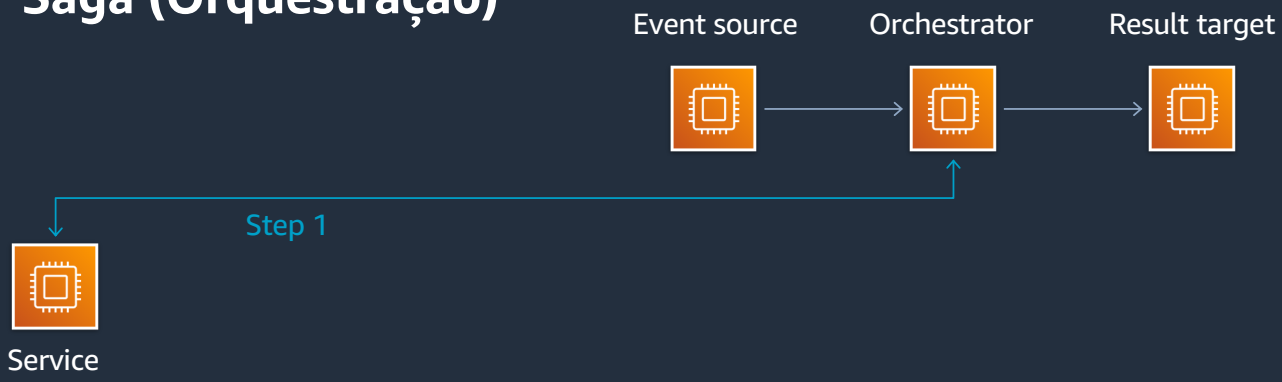
Result target

Pattern aplicado a Transição de Estado

Saga (Orquestração)

Pattern aplicado a Transição de Estado

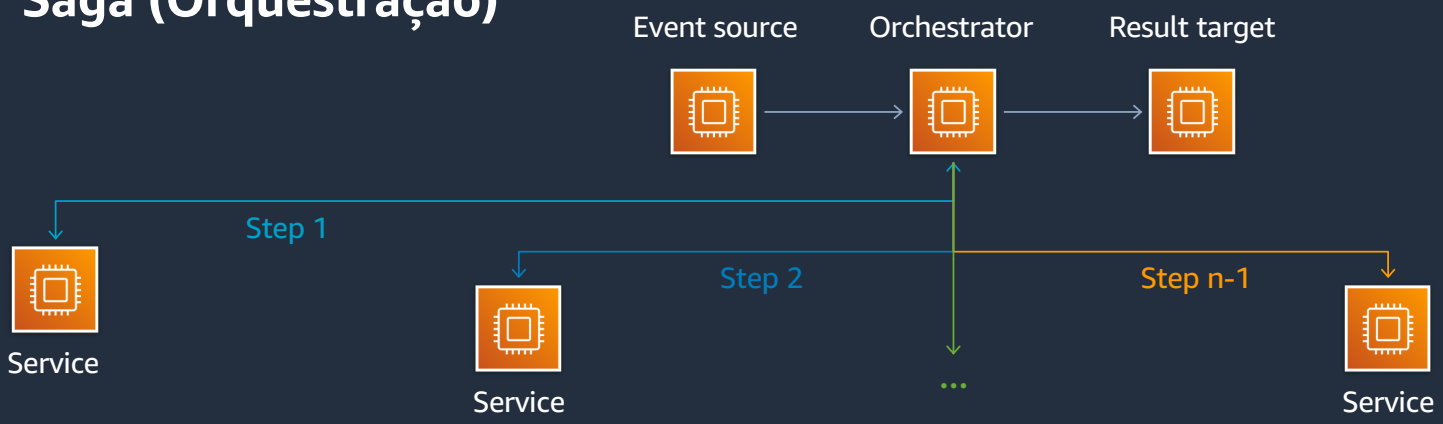
Saga (Orquestração)



Lógica externalizado pelo orquestrador (Saga Execution Coordinator)

Pattern aplicado a Transição de Estado

Saga (Orquestração)

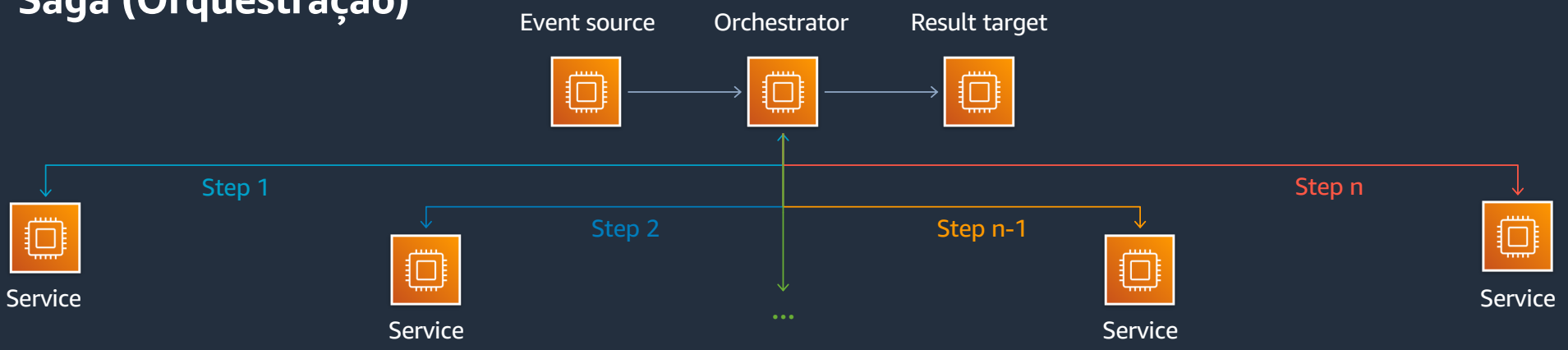


Lógica externalizado pelo orquestrador (Saga Execution Coordinator)

Participantes permanecem desacoplados mesmo em cenários de transações de longa duração

Pattern aplicado a Transição de Estado

Saga (Orquestração)



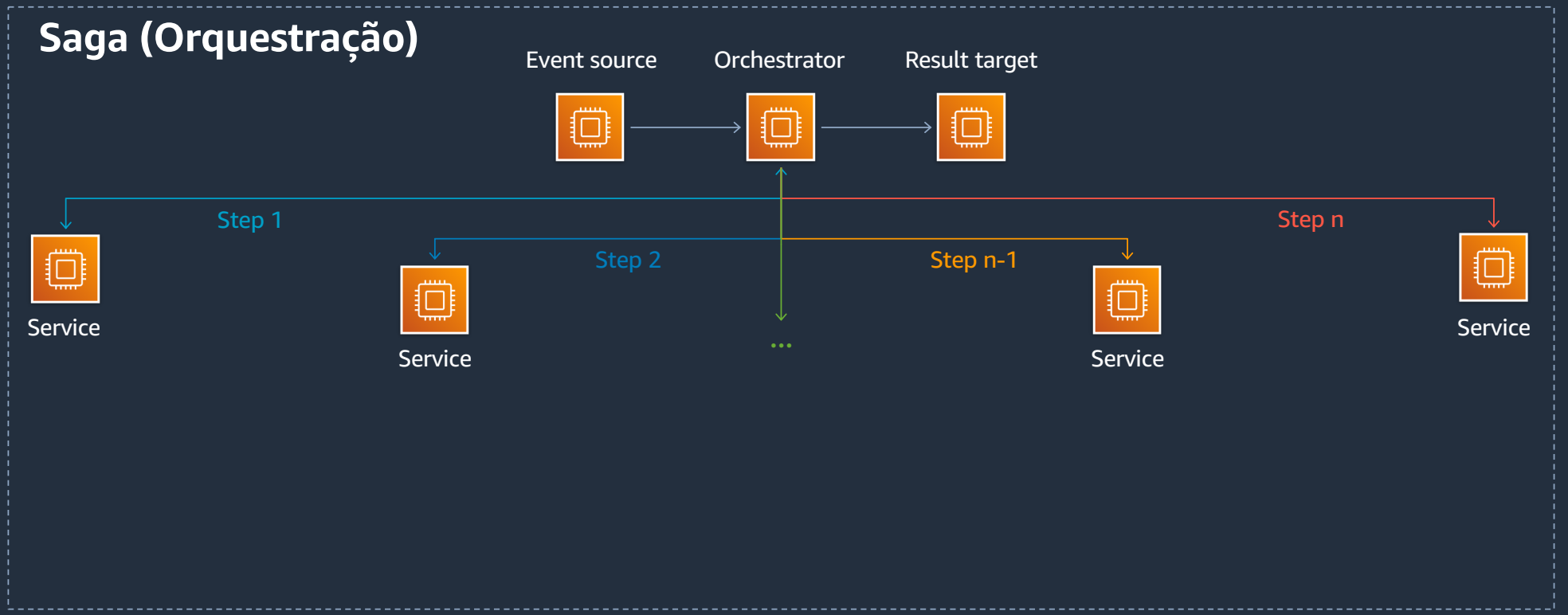
Lógica externalizado pelo orquestrador (Saga Execution Coordinator)

Participantes permanecem desacoplados mesmo em cenários de transações de longa duração

Falhas são tratadas usando o conceito de Compensation

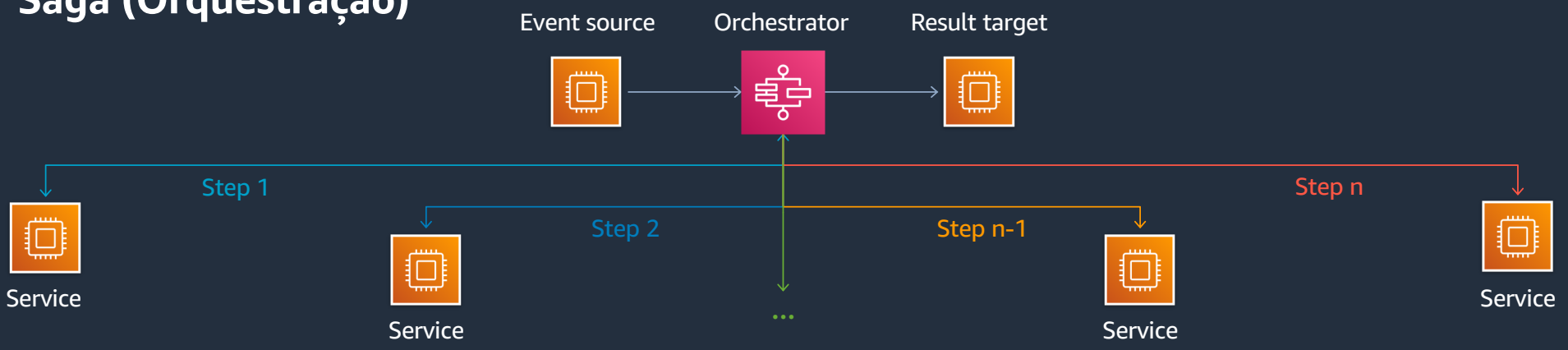

...quero ver FUNCIONANDO...

Pattern aplicado a Transição de Estado



Pattern aplicado a Transição de Estado

Saga (Orquestração)



Serviço AWS para saga orchestration (serverless) >> AWS Step Functions

AWS Step Functions

Serviço gerenciado de máquina de estados na AWS

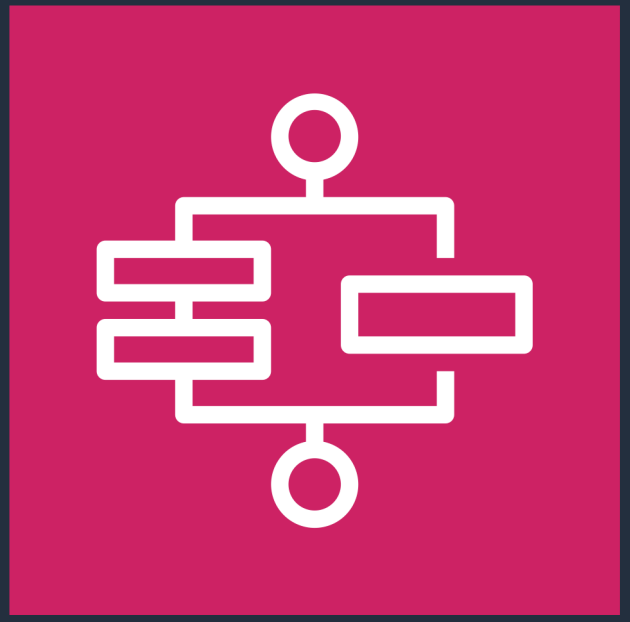
Automação de workflow resiliente

Suporte a manipulação de erros (Built-in)

Integração nativa com serviços AWS

Alta disponibilidade (HA) & escalar automaticamente

Histórico de execução auditável



aws

Caso de Uso: Triagem de pessoas infectadas pelo COVID-19

Fonte: Bloomberg / Getty Images

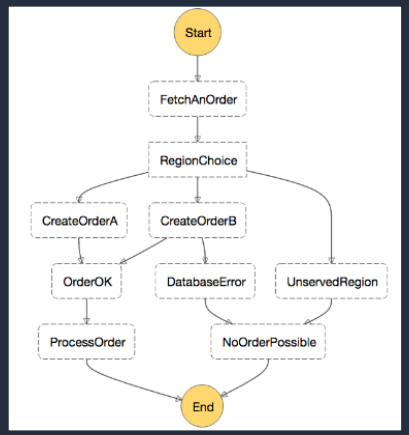
aws

Operando com o AWS Step Functions

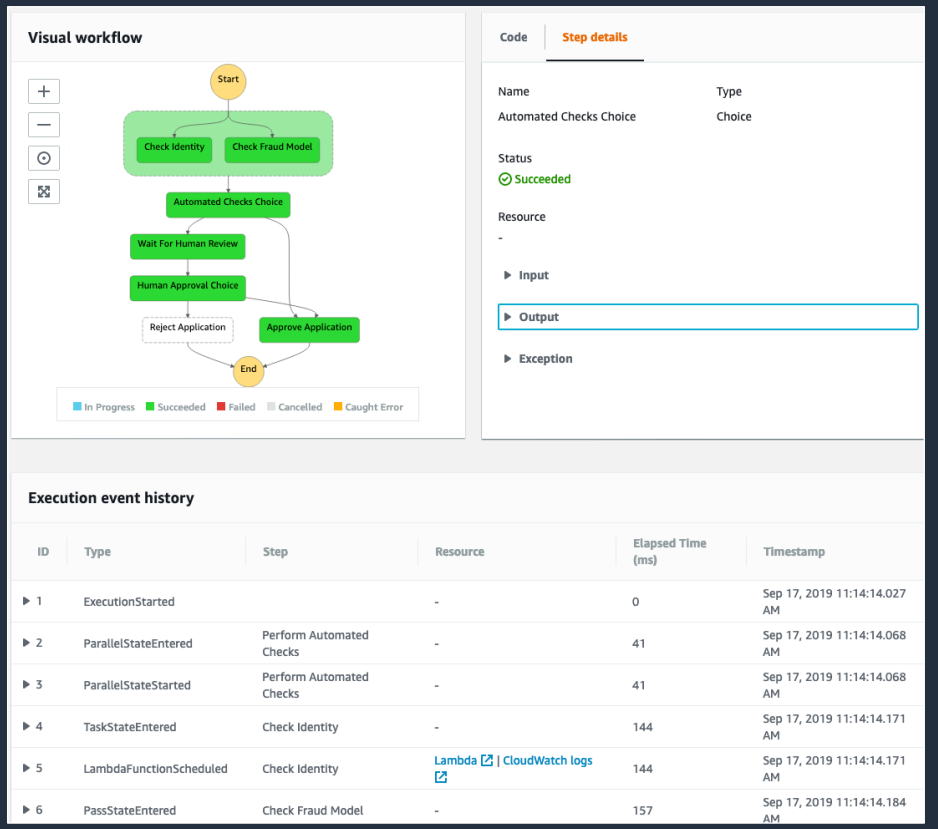
Definição em JSON

```
{
    "Comment": "Manage opening an account",
    "StartAt": "Perform Automated Checks",
    "States": {
        "Perform Automated Checks": {
            "Type": "Parallel",
            "Branches": [{
                    "StartAt": "Check Identity",
                    "States": {
                        "Check Identity": {
                            "Type": "Task",
                            "Parameters": {
```

Visualizar a Console



Monitorar Execuções



Amazon States Language

https://states-language.net/spec.html

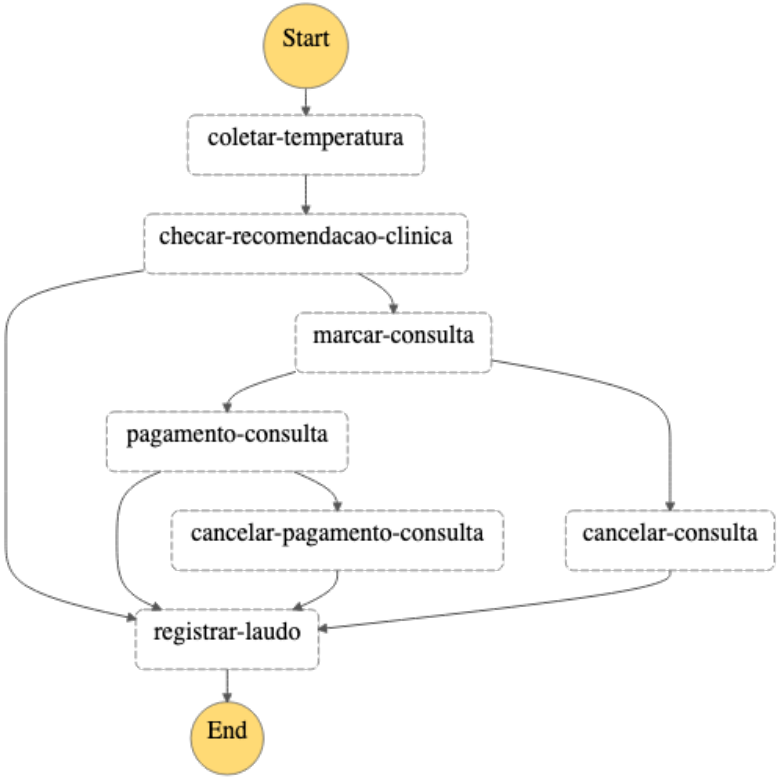
```
{
   "Comment": ”Verificar coleta de temperatura",
   "StartAt": ”coletar-temperatura",

   "States": {
    " coletar-temperatura ": {
         "Type": "Task",
         "Resource": "arn:aws:lambda...Triagem",
         "End": true
    },
    [. . .]
  }
}
```

aws

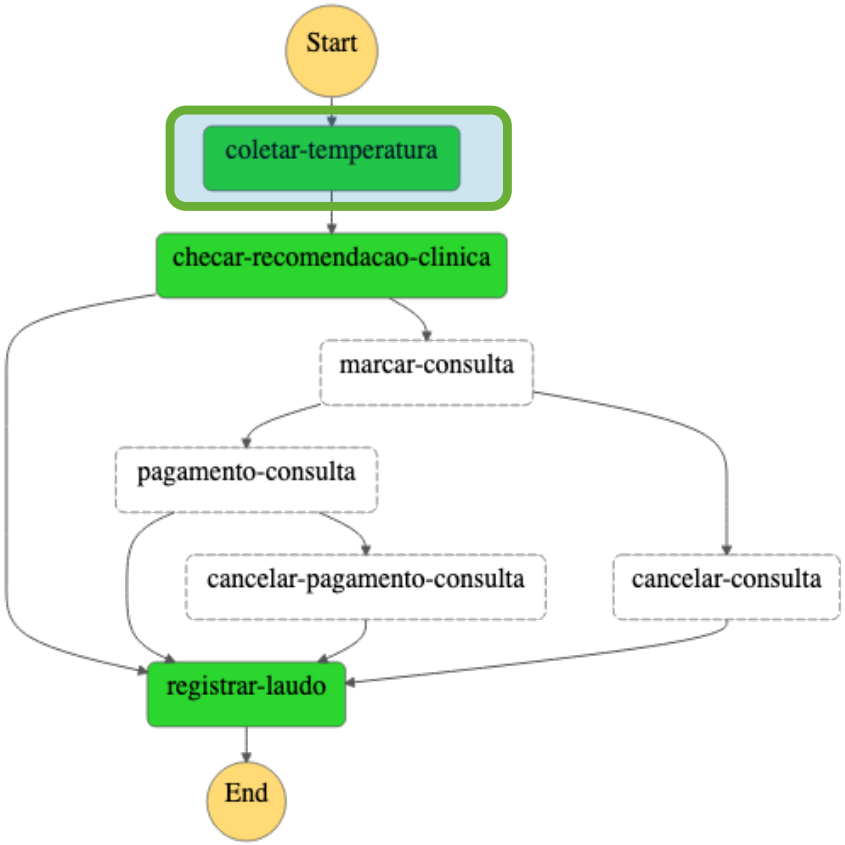
Adição de Lógica a Transição de Estado

- Tasks
- Parallel Steps
- Branching Choice
- Wait for a callback



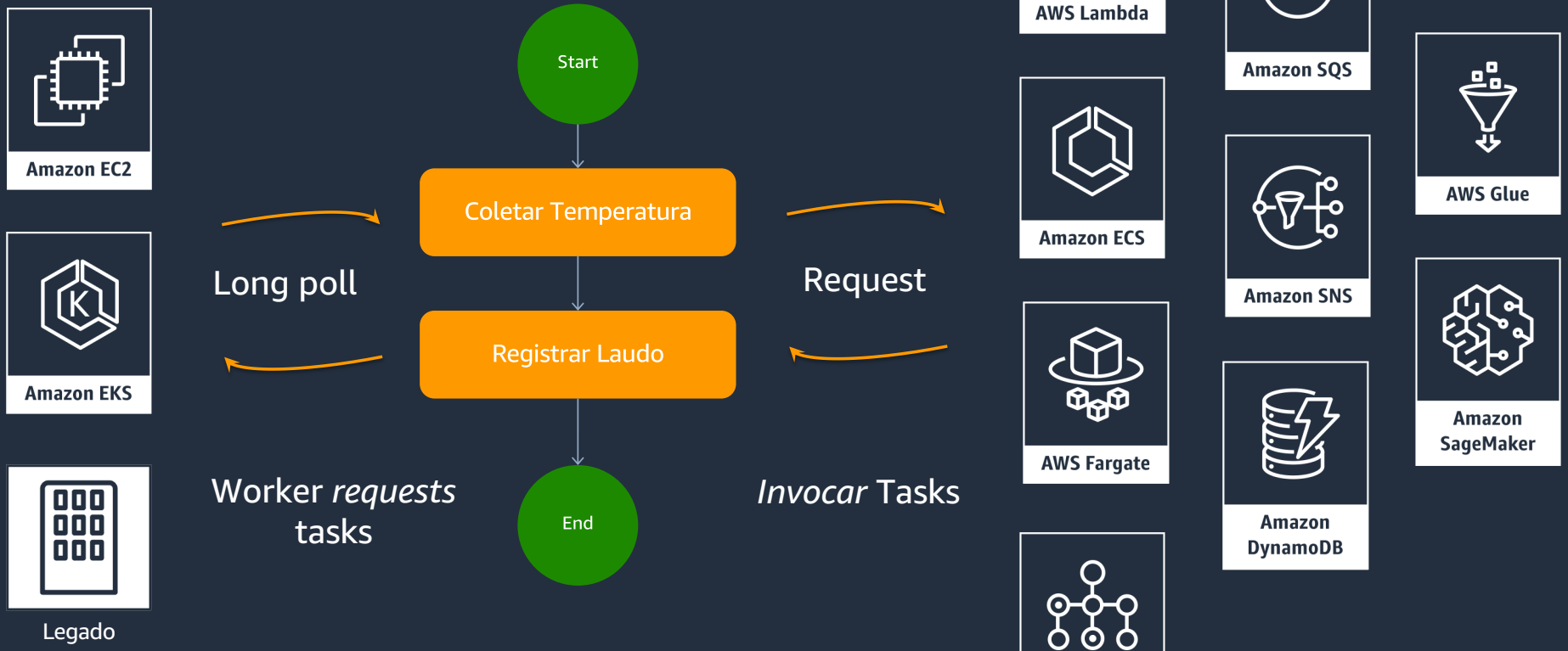
Executando uma Tarefa

```
"coletar-temperatura": {
    "Type": "Task",
    "Resource": "arn:aws:lambda...triagem",
    "Next": "checar-recomendacao-clinica"
}
```




Que tal dar uma turbinada
neste exemplo

Integração com outros serviços na nuvem



Automação. Escolha o que melhor adequa-se a sua realidade

AWS CloudFormation

SAM

CDK

CHALICE

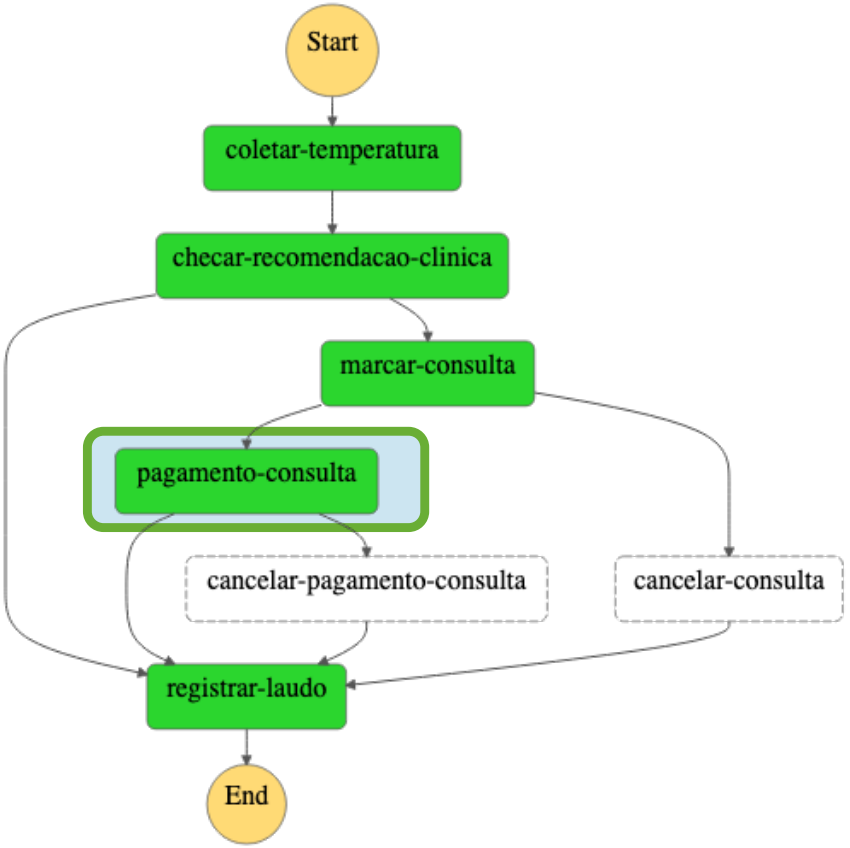
serverless

HashiCorp Terraform

aws

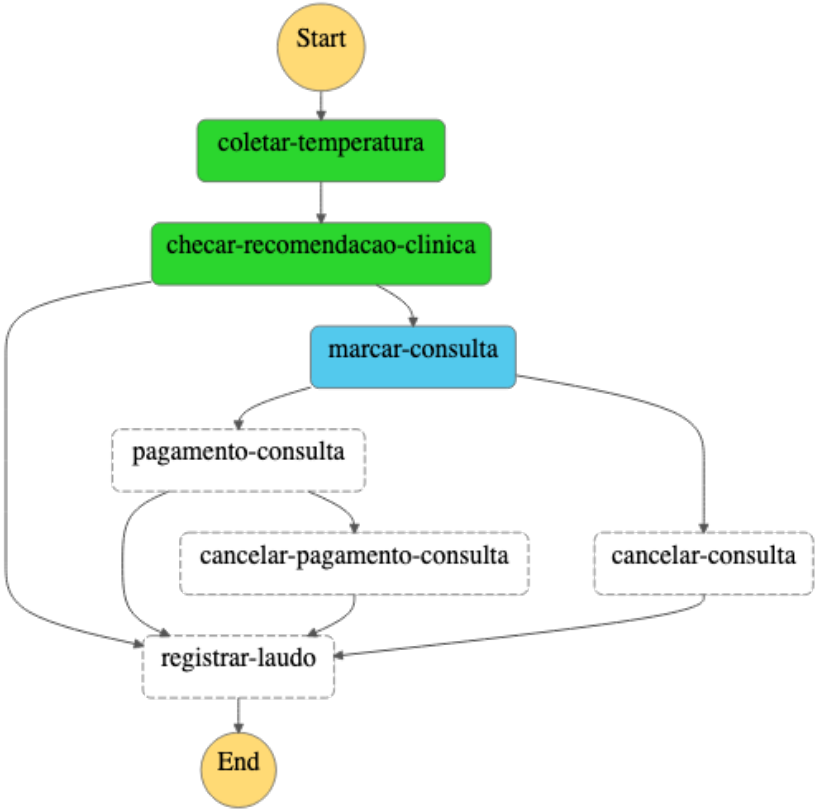
Uso de Retry e Compensation

```
"marcar-consulta": {
    "Type": "Task",
    "Resource": "${MarcarConsultaFunctionArn}",
    "ResultPath": "$.laudo",
    "Next": "pagamento-consulta",
    "Retry": [
      {
        "ErrorEquals": [
          "States.ALL"
        ],
        "IntervalSeconds": 1,
        "MaxAttempts": 3
      }
    ],
    "Catch": [ {
        "ErrorEquals": ["States.All"],
        "Next": "cancelar-consulta"
    }]
```



Aguardando callback no SQS

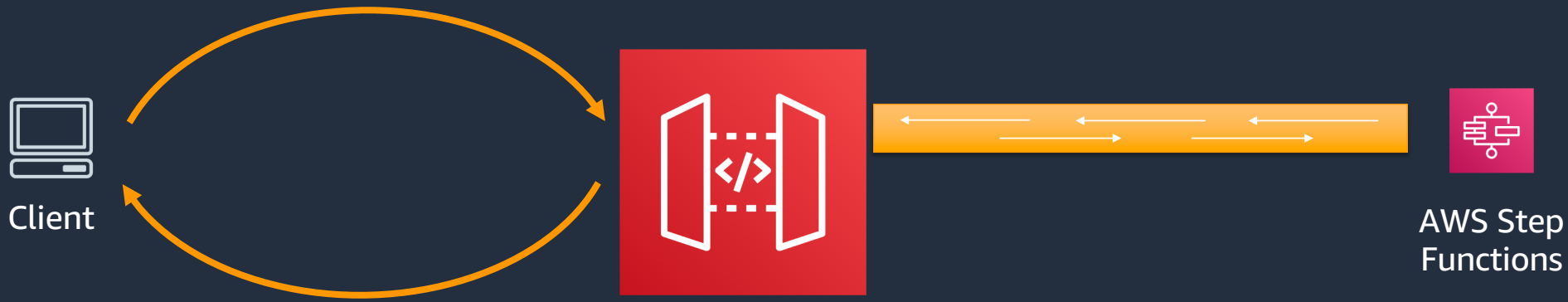
```
"Type": "Task",
"Resource":"arn:aws:states:::sqs:sendMessage.waitForTaskToken",
"Parameters": {
    ”QueueUrl": "https://sqs.us-east-1.amazonaws.com/123/sqsqueue",
    ”MessageBody": {
        ”Paciente": "$.paciente-id",
        ”TaskToken.$": "$$.Task.Token"
    }
},
"ResultPath": "$.laudo",
"Next": " pagamento-consulta"
```



E no final tudo vira API com API Gateway ...

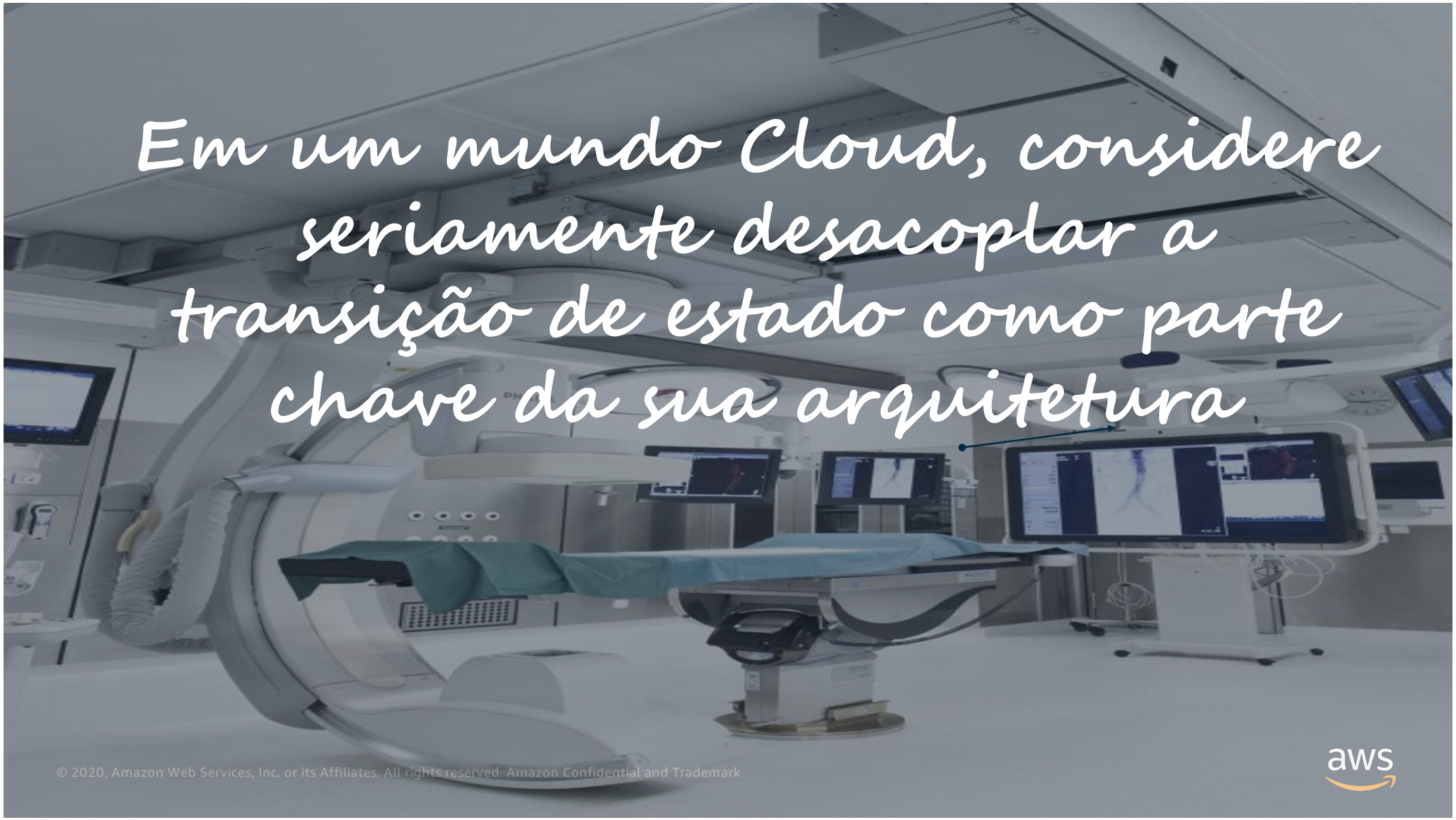
RESTful: HTTP APIs & REST APIs

WebSocket APIs



- Request / Response
- HTTP Methods like GET, POST, etc
- Short-lived communication
- Stateless

- Serverless WebSocket
- 2 way communication channel
- Long-lived communication
- Stateful



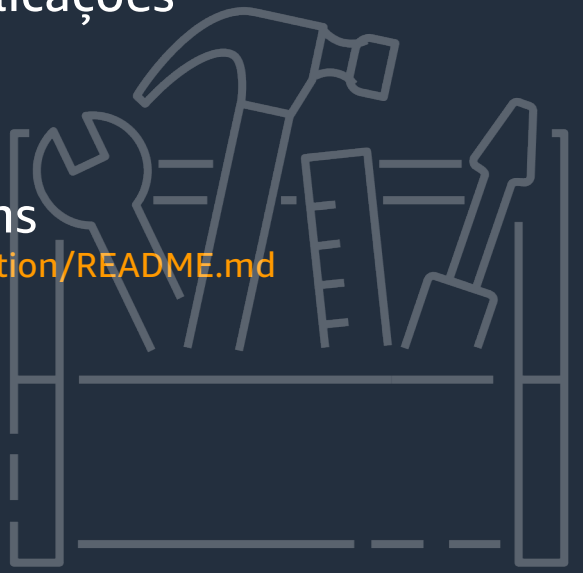
Dicas

AWS Step Functions
https://docs.aws.amazon.com/step-functions/latest/dg/sfn-local.html

AWS blogs e outros conteúdos para integração de aplicações
https://bit.ly/aws-msgn

Automação de Saga Pattern com CDK e Step Functions
https://github.com/cdk-patterns/serverless/blob/master/the-saga-stepfunction/README.md

Visual Studio Code - AWS Toolkit extension
https://aws.amazon.com/visualstudiocode/



Muito Obrigado

Alex Coqueiro
Head de Arquitetura de Soluções para América Latina, Canada e Caribe
AWS Public Sector
@alexbcbr