MBA EM **ENGENHARIA DE SOFTWARE**

Serverless II

Prof. José Maria Cesário Jr.

MBA USP ESALQ

A responsabilidade pela idoneidade, originalidade
e licitude dos conteúdos didáticos apresentados é do professor.

MBA USP ESALQ

José Maria Cesário Júnior

- José Maria Cesário Júnior possui mais de 25 anos de experiência na área de TI e atua como Arquiteto de Soluções para Parceiros na AWS Brasil.
- Possui experiência em soluções de Cloud Computing, Indústria 4.0 e IoT Industrial
- Atua desde 2007 como professor universitário de Graduação e Pós Graduação.
- Analista de Sistemas, graduado pela Universidade Metodista de Piracicaba, com especialização em Tecnologia
- MBA em Gestão de Projetos pela FGV
- Especialista em Automação e Controle de Processos Industriais e Agroindustriais pela FEAGRI, UNICAMP
- Mestre em Tecnologia e Inovação pela UNICAMP

https://www.linkedin.com/in/josemariacesariojunior/

AGENDA

Data	Conteúdo
25/09/2025	• Padrões de Arquitetura Serverless • Processamento Síncrono e Assíncrono • Casos de Uso Orientados por APIs • Aplicações síncronas • Pensando de forma assíncrona • Casos de Uso Orientados por Eventos (Event-Driven) • Aplicações assíncronas • Encerramento

ÍCONES

DEFINIÇÃO OU PONTO DE ATENÇÃO

EXERCÍCIO OU ATIVIDADE PRÁTICA

ACESSO AO AMBIENTE NA NUVEM

! IMPORTANTE

O laboratório prático utilizará recursos da AWS (Amazon Web Services), Microsoft Azure ou GCP (Google Cloud Platform) que podem gerar custos reais. Pontos essenciais:

1. A participação nesta atividade é OPCIONAL
2. Cada aluno é responsável pelo uso de sua própria conta no provedor de serviços Cloud
3. Os provedores de serviços Cloud oferecem um nível gratuito (Free Tier) com limites específicos
4. Custos além do Free Tier serão de responsabilidade individual do usuário
5. Recomendamos verificar os limites do Free Tier antes de iniciar as atividades práticas em https://aws.amazon.com/pt/free/ , https://azure.microsoft.com/pt-br/pricing/free-services, https://cloud.google.com/free/docs/free-cloud-features

Referências e Fontes

- Esse material tomou como referência a documentação oficial e aberta dos principais provedores e fabricantes de serviços de computação em nuvem pública do mercado
- AWS Amazon Web Services
- GCP Google Cloud Platform
- Microsoft Azure

Referências Fabricantes

https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/introduction.html

https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/welcome.html

https://serverlessland.com/

Referências Fabricantes

Google Cloud

https://cloud.google.com/architecture/blueprints/serverless-blueprint

https://cloud.google.com/architecture/framework

https://cloud.google.com/serverless?hl=en

Referências Fabricantes

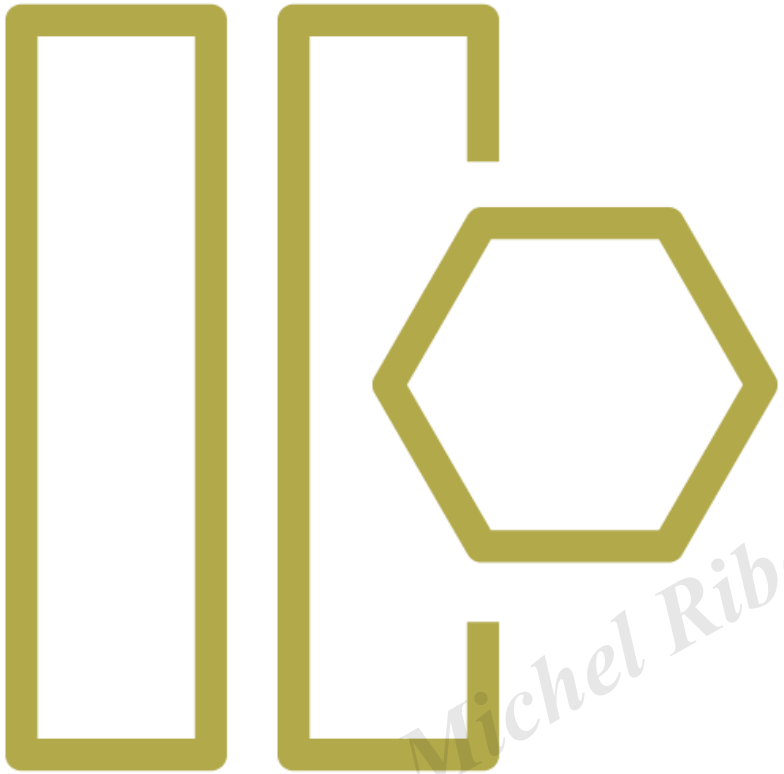
https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-serverless-computing

https://azure.microsoft.com/en-us/solutions/serverless

https://azure.microsoft.com/en-us/products/functions/

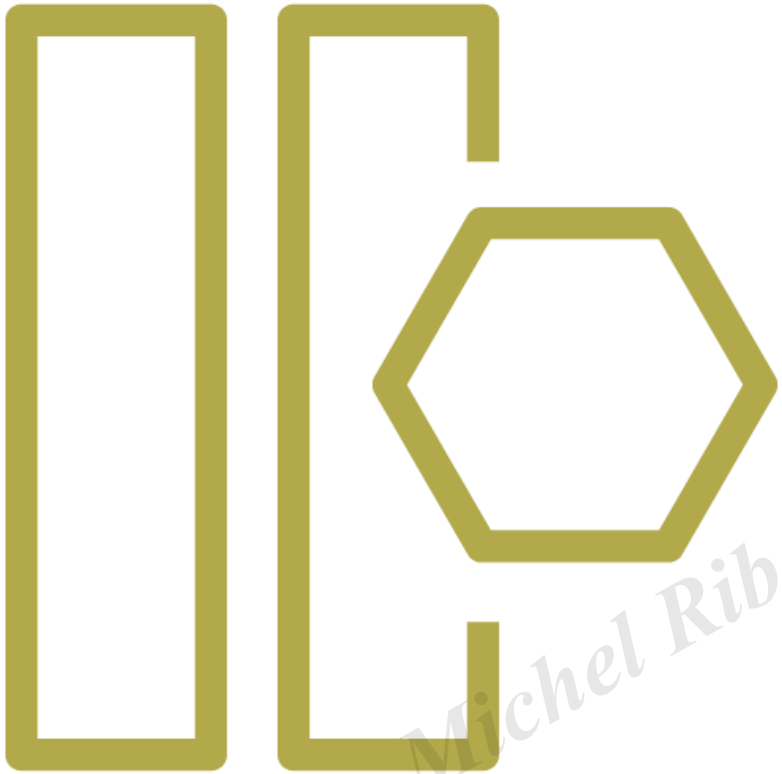
Padrões de Arquiteturas Serverless

PARTE 1



evento

Uma coisa que acontece, particularmente uma coisa importante.



evento

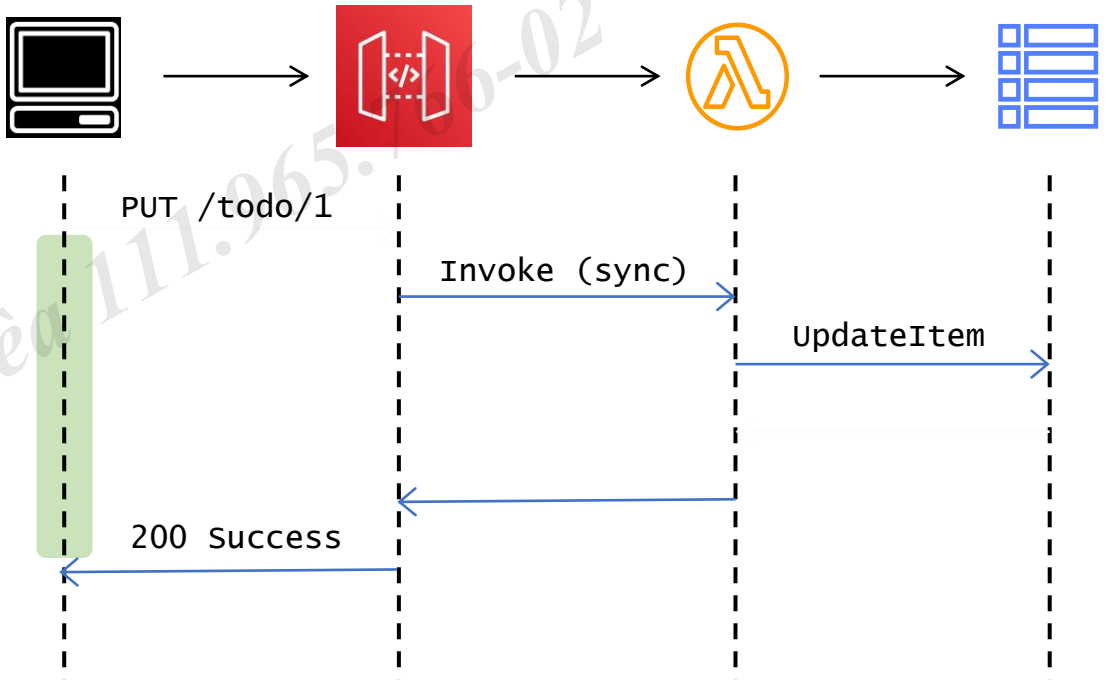
Em arquiteturas serverless

Tecido de sistemas fracamente acoplados.

Representa a mudança no estado, melhorando a resiliência.

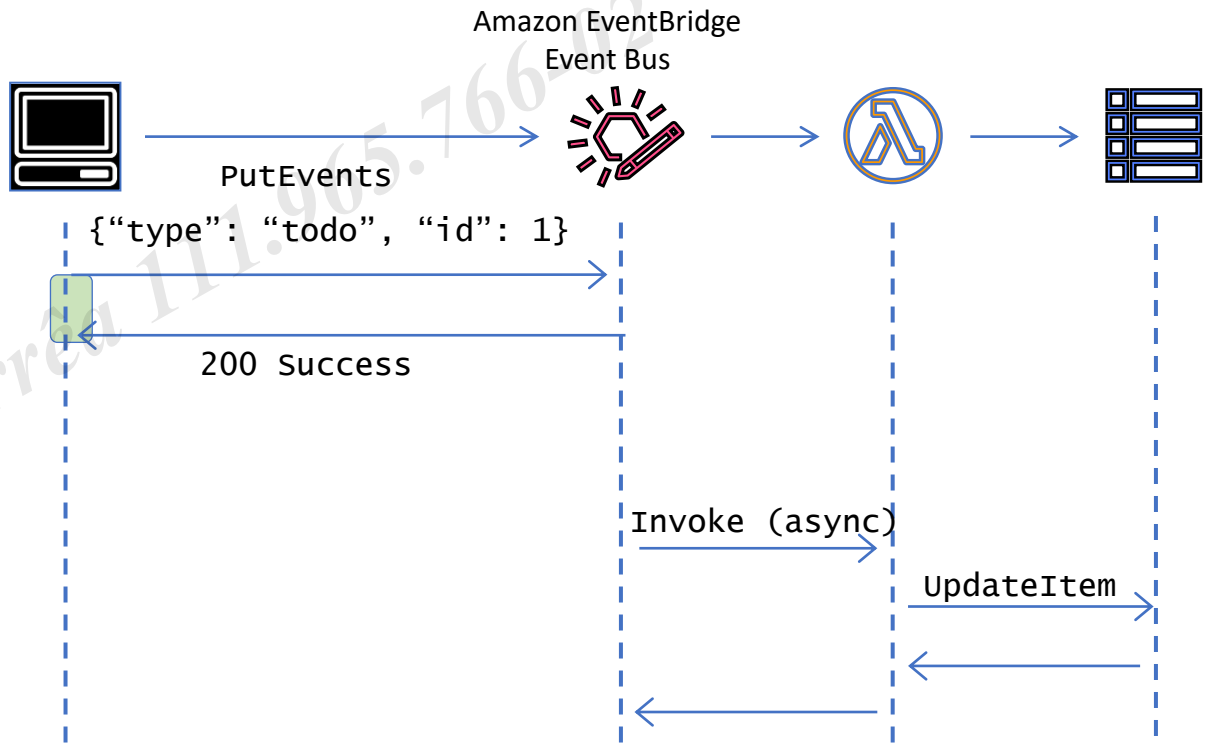
Arquiteturas orientadas por API

- A API define a interface
  - por exemplo, REST, GraphQL
- O chamador espera uma resposta imediata
  - A resposta contém o resultado do trabalho
  - Geralmente, menos de 30 segundos
  - Processamento síncrono
- O cliente deve implementar o tratamento de erros e a lógica de repetição

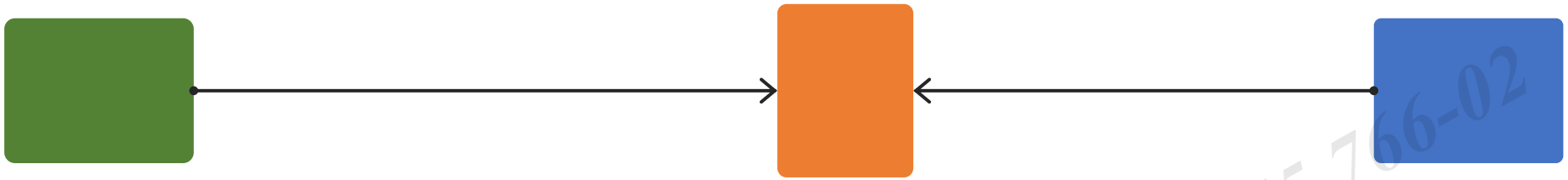


Arquiteturas orientadas por eventos
(EDA: Event-Driven Architectures)

- A carga útil do evento define a interface
  - Por exemplo, JSON
- A resposta é “mensagem recebida” (ou não)
  - Processamento assíncrono
  - O processo pode ser de longa duração (além dos 30s)
  - Atualizar o cliente (por exemplo, UI) pode ser mais desafiador
- Maior resiliência e durabilidade
  - Impulsionado pelo serviço de mensagens
- Tentativas integradas, configuráveis para o caso de uso



Componentes EDA



Produtor

- Os produtores de eventos são sistemas que detectam uma mudança no estado ou notificam atualizações e publicam esses fatos.
- O código do aplicativo, um banco de dados ou um acionador baseado em tempo podem servir como produtores de eventos, entre outras coisas.

Roteador

- Um roteador de eventos é um ponto de encontro entre produtores e consumidores.
- O roteador define como os eventos são trocados.
- Também conhecido como “event-bus”

Consumidor

- Os consumidores de eventos são sistemas que escutam eventos e os usam para seus propósitos.
- É possível e comum que um consumidor receba um evento, realize algum trabalho e publique seu evento em resposta.

Fonte: Adaptado de https://aws.amazon.com/pt/event-driven-architecture/

Roteadores de eventos

Pull consulta eventos periodicamente
Push envia eventos ativamente

Baseado em pull

Filas

Amazon SQS

Amazon SNS

Amazon MQ

Fluxos

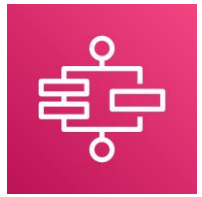
Amazon Kinesis

Amazon Managed Streaming for Apache Kafka

Baseado em push

Roteadores

Amazon EventBridge



AWS Step Functions

Processamento Síncrono e Assíncrono

PARTE 2

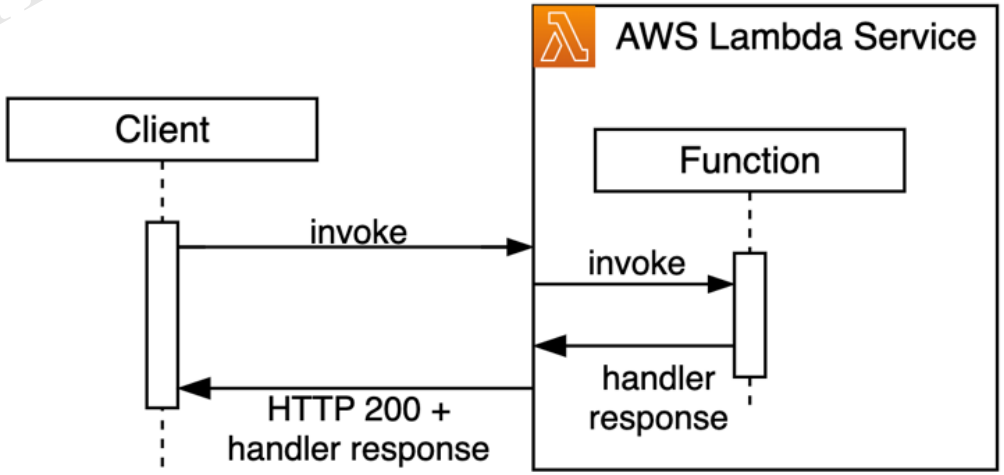
Modo síncrono ou assíncrono

- Uma função do AWS Lambda pode ser chamada tanto de forma síncrona quanto assíncrona, dependendo dos requisitos da aplicação e do serviço que a invoca.
- Na invocação síncrona, o cliente aguarda a resposta da função
- Na invocação assíncrona, o Lambda coloca o evento numa fila e retorna uma resposta imediatamente, sem esperar pela conclusão da função.

Fonte: Adaptado de https://aws.amazon.com/pt/blogs/aws-brasil/lidando-com-bilhoes-de-invocacoes-melhores-praticas-do-aws-lambda/

Modo síncrono

- As invocações síncronas são normalmente usadas para aplicativos interativos que esperam respostas imediatas, como APIs da web.
- O serviço Lambda recebe a solicitação de invocação, invoca o manipulador (handler) da função, espera pela resposta do manipulador e a retorna em resposta à solicitação original.
- Com as invocações síncronas, o cliente espera o retorno do manipulador de funções e é responsável por gerenciar os tempos limite e as novas tentativas de invocações com falha.

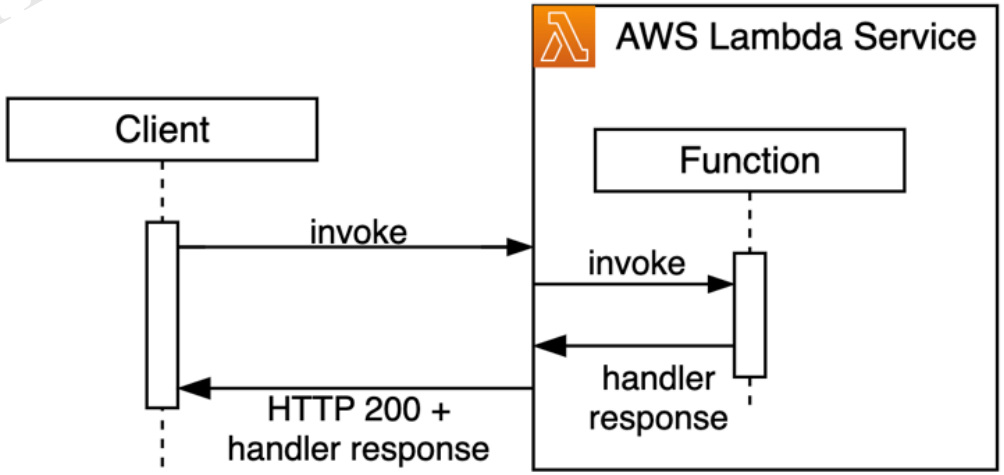


Fonte: Adaptado de https://aws.amazon.com/pt/blogs/aws-brasil/lidando-com-bilhoes-de-invocacoes-melhores-praticas-do-aws-lambda/

Modo assíncrono

- As invocações assíncronas permitem execuções de funções desacopladas. Os clientes enviam cargas para processamento sem esperar respostas imediatas. Isso é usado para cenários como processamento assíncrono de dados ou envios de pedidos/trabalhos.

- O serviço Lambda retorna imediatamente uma confirmação da invocação aceita e continua gerenciando outras invocações, tempos limite e novas tentativas do manipulador (handler) de forma assíncrona.

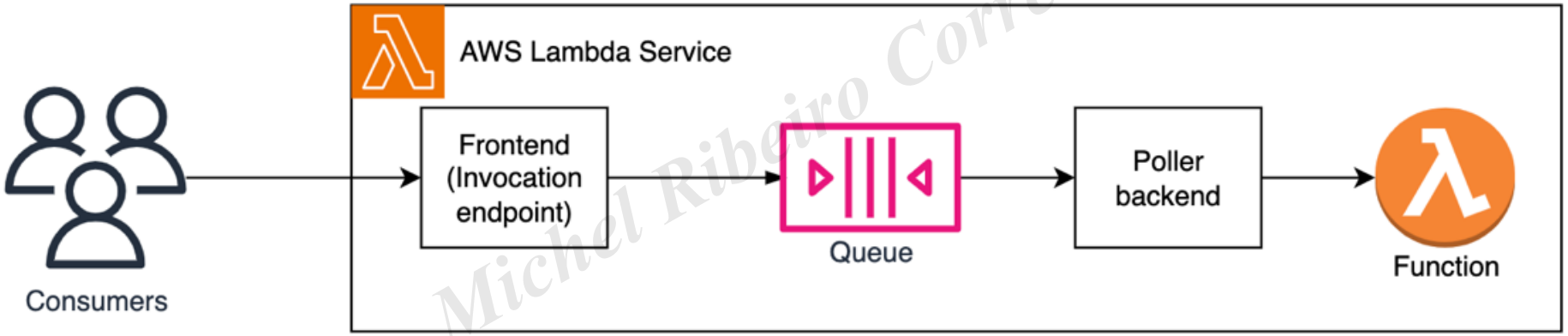


Fonte: Adaptado de https://aws.amazon.com/pt/blogs/aws-brasil/lidando-com-bilhoes-de-invocacoes-melhores-praticas-do-aws-lambda /

Modo assíncrono

- Para acomodar invocações assíncronas, o serviço Lambda coloca as solicitações em sua fila interna e retorna imediatamente o HTTP 202 ao cliente.

- Depois disso, um componente de pesquisa interno separado lê as mensagens da fila e invoca a função de forma síncrona.



Fonte: Adaptado de https://aws.amazon.com/pt/blogs/aws-brasil/lidando-com-bilhoes-de-invocacoes-melhores-praticas-do-aws-lambda/

Modo assíncrono: falhas e erros

- O Lambda gerencia a fila de eventos assíncronos da função e realiza novas tentativas em caso de erro.

- Se a função retornar um erro, o Lambda, por padrão, tentará executá-la mais duas vezes, com um intervalo de um minuto entre a primeira e a segunda tentativa, e um intervalo de dois minutos entre a segunda e a terceira tentativa.

- Pode-se utilizar a Dead Letter Queue para armazenar eventos que falharam após todas as tentativas de retry, permitindo análise posterior de falhas sem perder dados

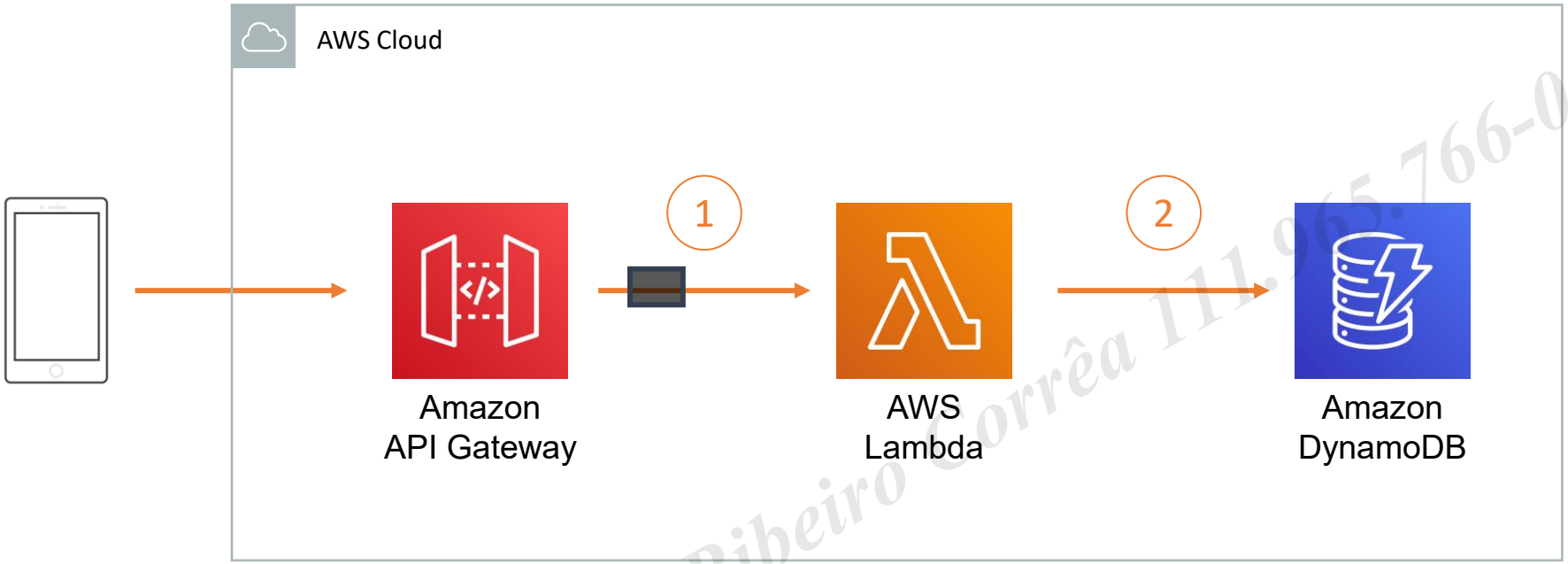
Fonte: Adaptado de https://docs.aws.amazon.com/pt_br/lambda/latest/dg/invocation-async-error-handling.html

Casos de Uso Orientados por APIs

Processamento Síncrono

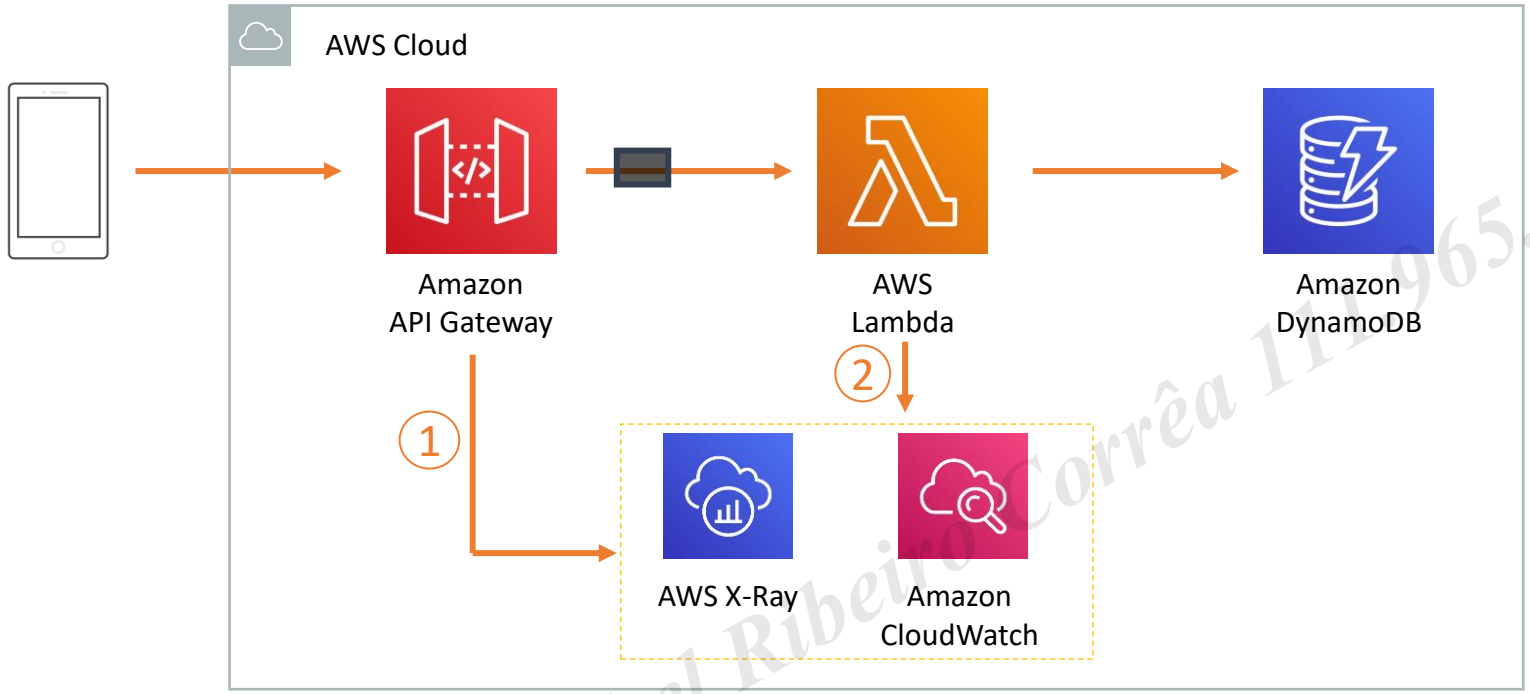
PARTE 3

Microsserviços REST



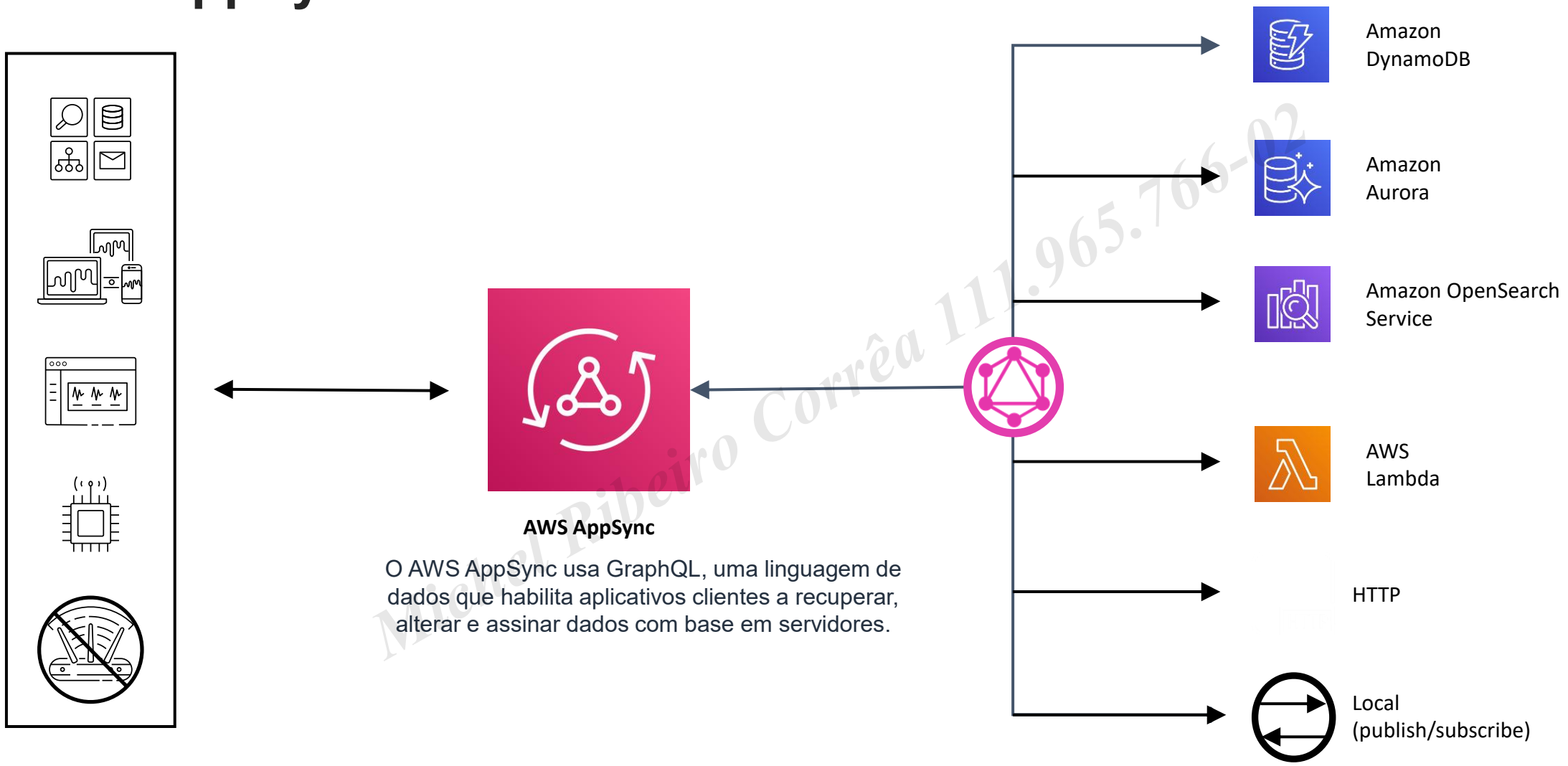
1. O API Gateway “traduz” a solicitação HTTP recebida na carga útil do evento
2. O Lambda lê e grava dados no banco de dados ou armazenamento

Microsserviços REST: observabilidade



1. Habilitar registros de acesso e rastreamento
2. Instrumentar o código e criar métricas de forma assíncrona com o CloudWatch Embedded Metric Format

AWS AppSync



AWS AppSync

O AWS AppSync usa GraphQL, uma linguagem de dados que habilita aplicativos clientes a recuperar, alterar e assinar dados com base em servidores.

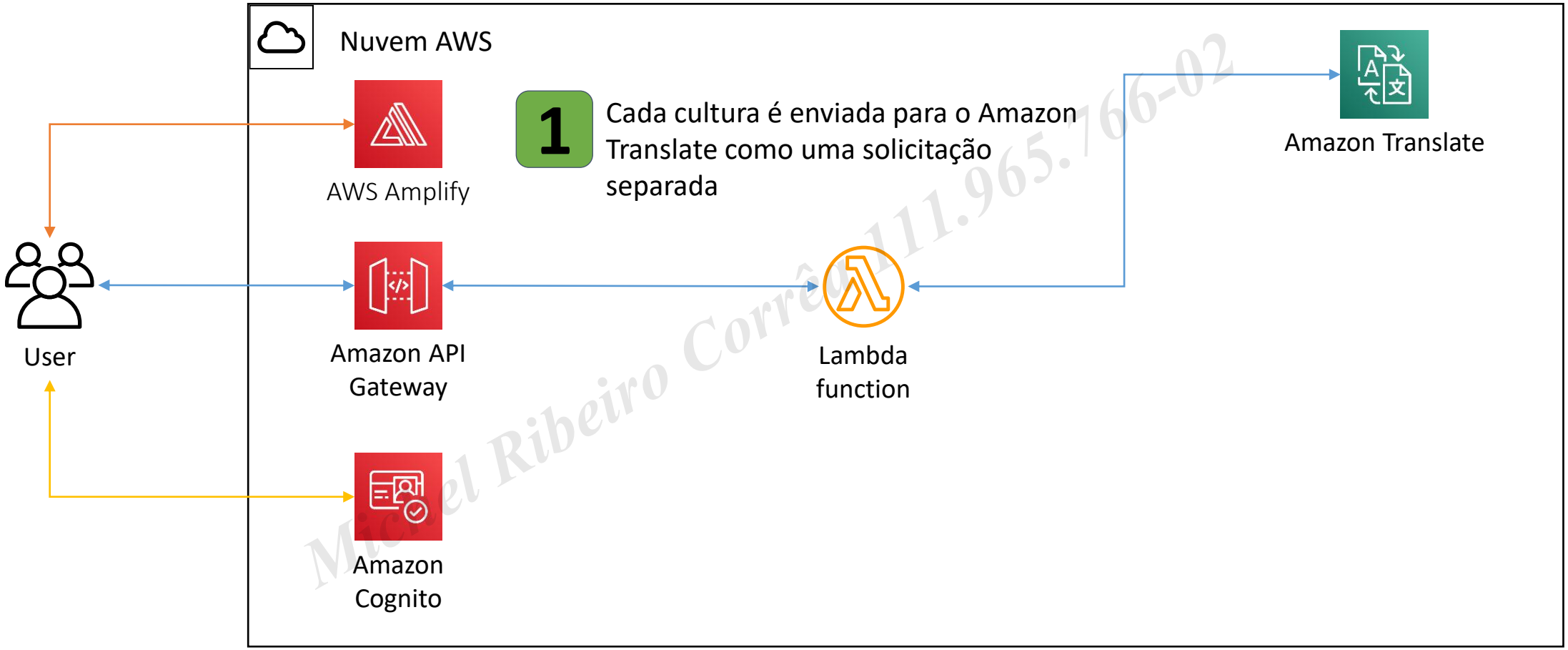
Aplicações síncronas

PARTE 4

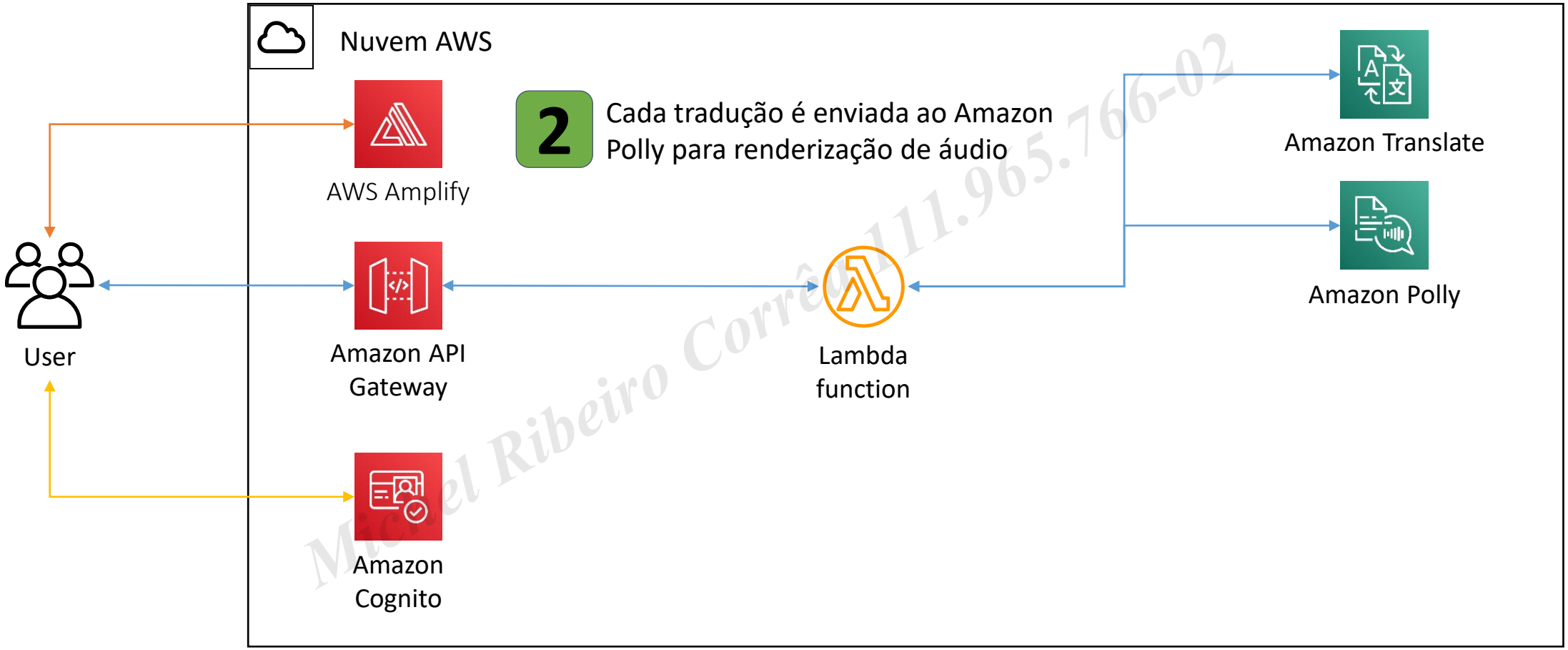
Dados do evento

```
{
      "data":”Hello, welcome to my translation application.",
      "culture":[”fr-FR","tr-TR”,“de-DE”,”fr-CA”]
}
```

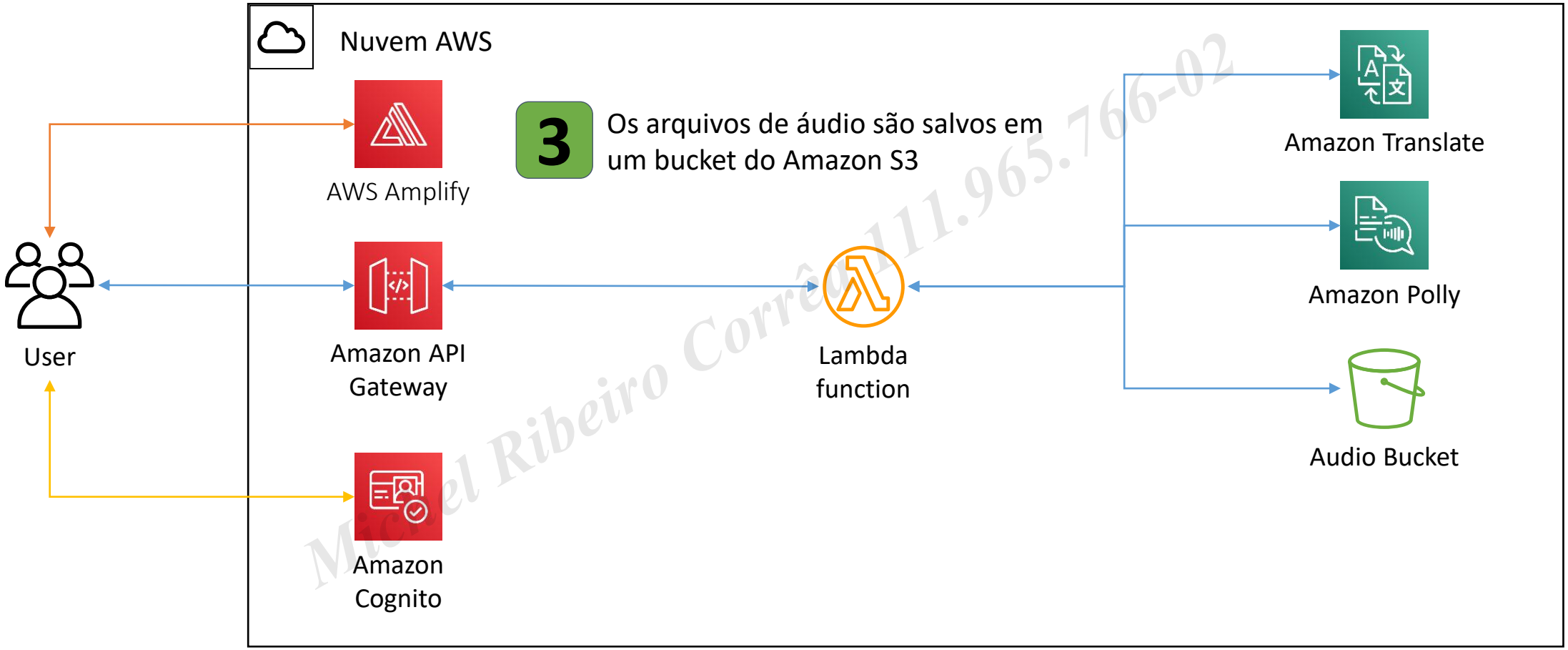
Exemplo de aplicativo síncrono



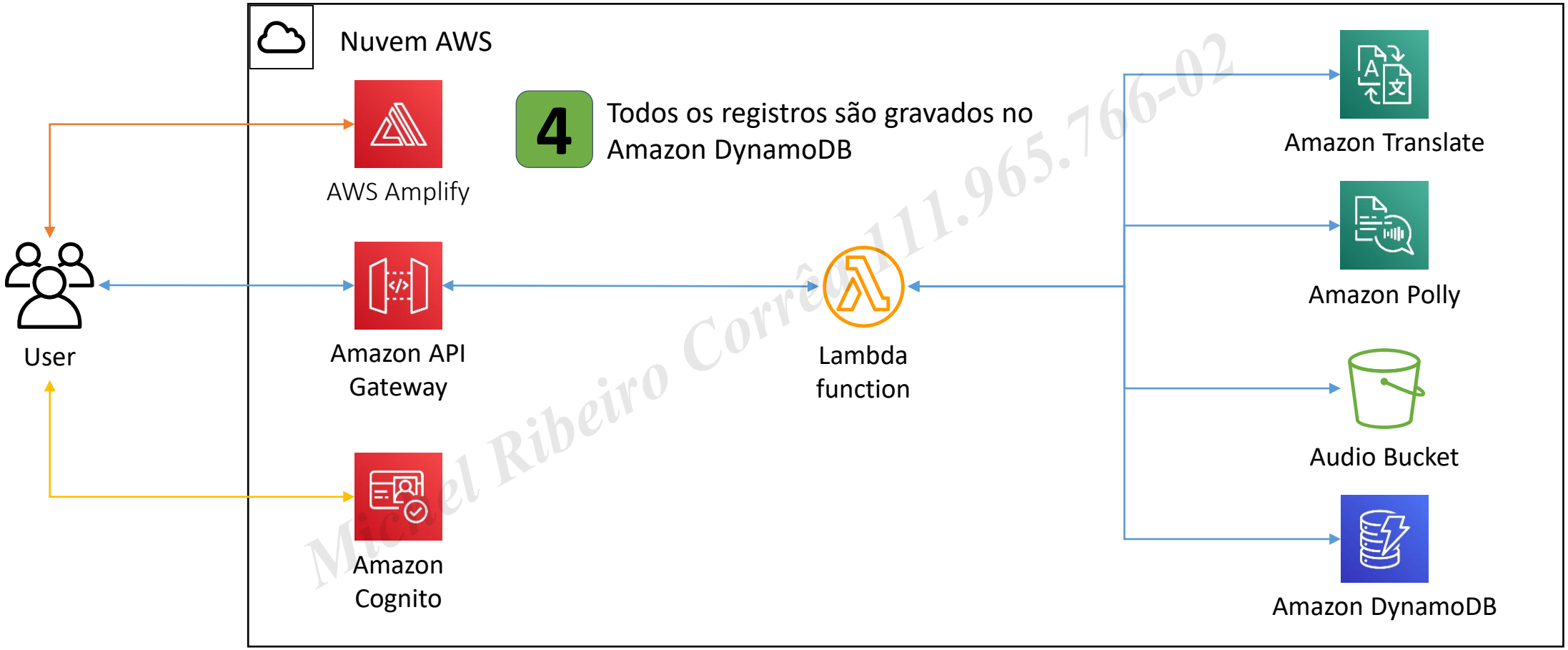
Exemplo de aplicativo síncrono



Exemplo de aplicativo síncrono



Exemplo de aplicativo síncrono



Dados resultantes

DynamoDB

id	type	culture	data	timestamp	originalId
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	audio-de-DE	de-DE	https://translator-synch-audiobucket-li8k1wox4fxm.s3-…	1568223241096	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	audio-fr-CA	fr-CA	https://translator-synch-audiobucket-li8k1wox4fxm.s3-…	1568223241096	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	audio-fr-FR	fr-FR	https://translator-synch-audiobucket-li8k1wox4fxm.s3-…	1568223241096	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	audio-tr-TR	tr-TR	https://translator-synch-audiobucket-li8k1wox4fxm.s3-…	1568223241096	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	original	[{ "S" : "fr-FR" }, { "S" : "tr-…	Hello, welcome to my translation demo.	1568223238116	original-55de7a40-d4ba-11e9-b999-e9c6d1f6a
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	translation-de-DE	de-DE	Hallo, willkommen zu meiner Übersetzungsdemo.	1568223239938	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	translation-fr-CA	fr-CA	Bonjour, bienvenue à ma démo de traduction.	1568223239938	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	translation-fr-FR	fr-FR	Bonjour, bienvenue à ma démo de traduction.	1568223239938	
55de7a40-d4ba-11e9-b999-e9c6d1f6a327	translation-tr-TR	tr-TR	Merhaba, çeviri demoma hoş geldiniz.	1568223239938	

S3

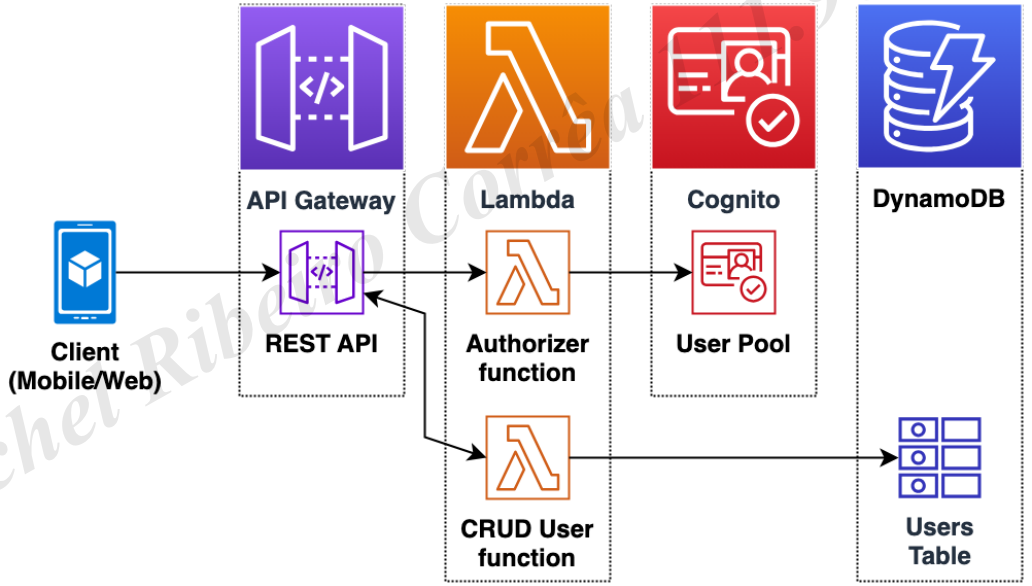
Name	Last modified	Size	Storage class
55de7a40-d4ba-11e9-b999-e9c6d1f6a327_de-DE_Marlene.mp3	Sep 11, 2019 11:34:01 AM GMT-0600	17.3 KB	Standard
55de7a40-d4ba-11e9-b999-e9c6d1f6a327_fr-CA_Chantal.mp3	Sep 11, 2019 11:34:02 AM GMT-0600	16.4 KB	Standard
55de7a40-d4ba-11e9-b999-e9c6d1f6a327_fr-FR_Celine.mp3	Sep 11, 2019 11:34:02 AM GMT-0600	16.6 KB	Standard
55de7a40-d4ba-11e9-b999-e9c6d1f6a327_tr-TR_Filiz.mp3	Sep 11, 2019 11:34:02 AM GMT-0600	15.8 KB	Standard

Viewing 1 to 4

Exercício

- Exemplo de aplicação síncrona

  https://catalog.workshops.aws/serverless-patterns/en-US/module2



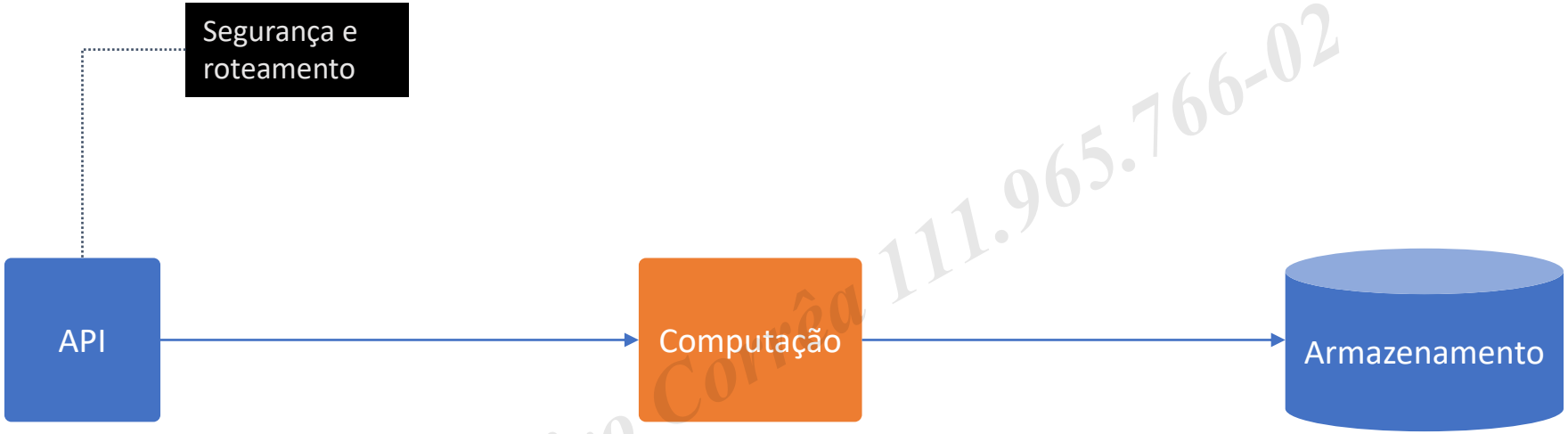
Fonte: https://catalog.workshops.aws/serverless-patterns/en-US/module2

Pensando de forma assíncrona

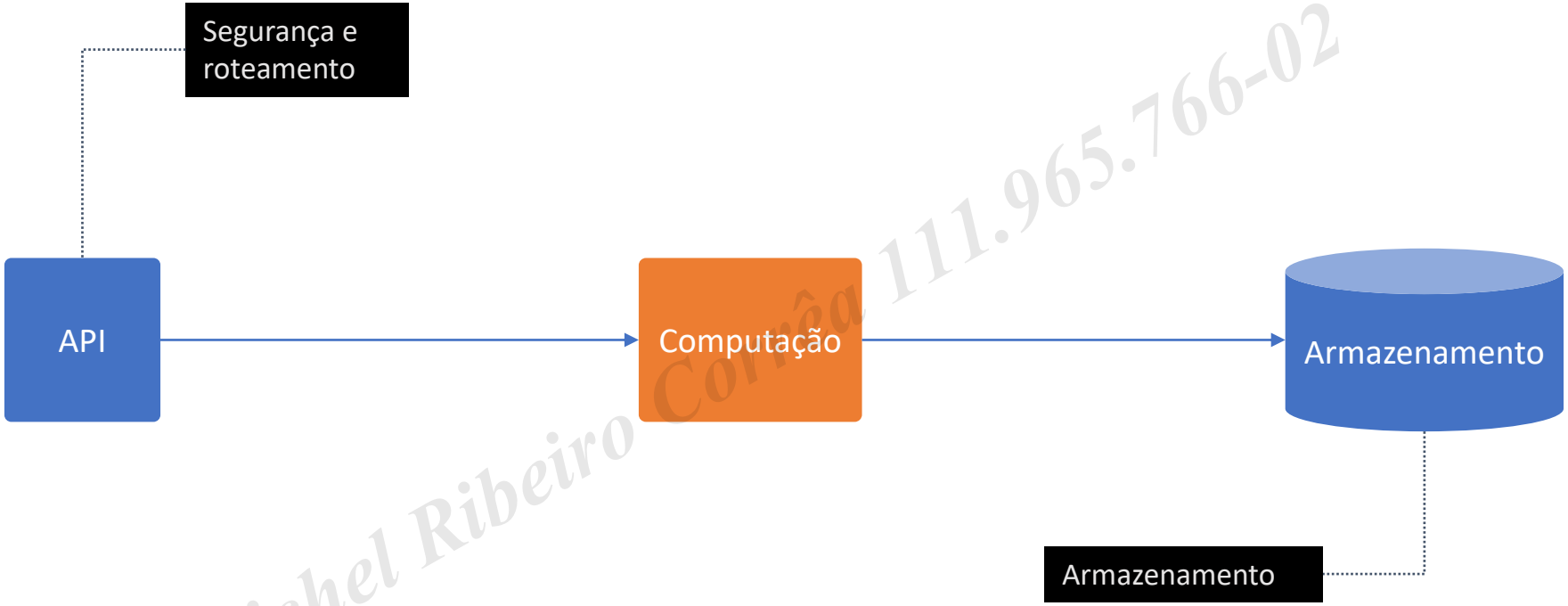
PARTE 5

Elementos do aplicativo

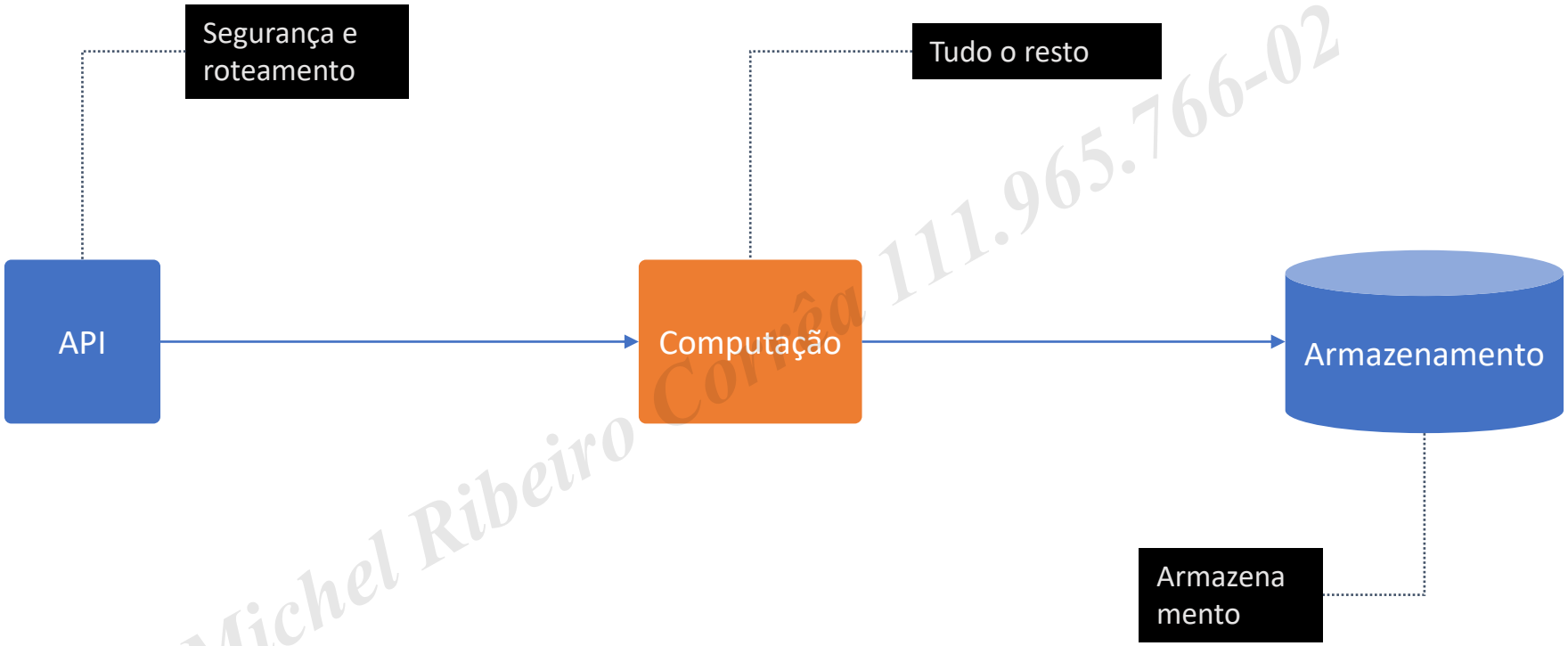
Elementos do aplicativo



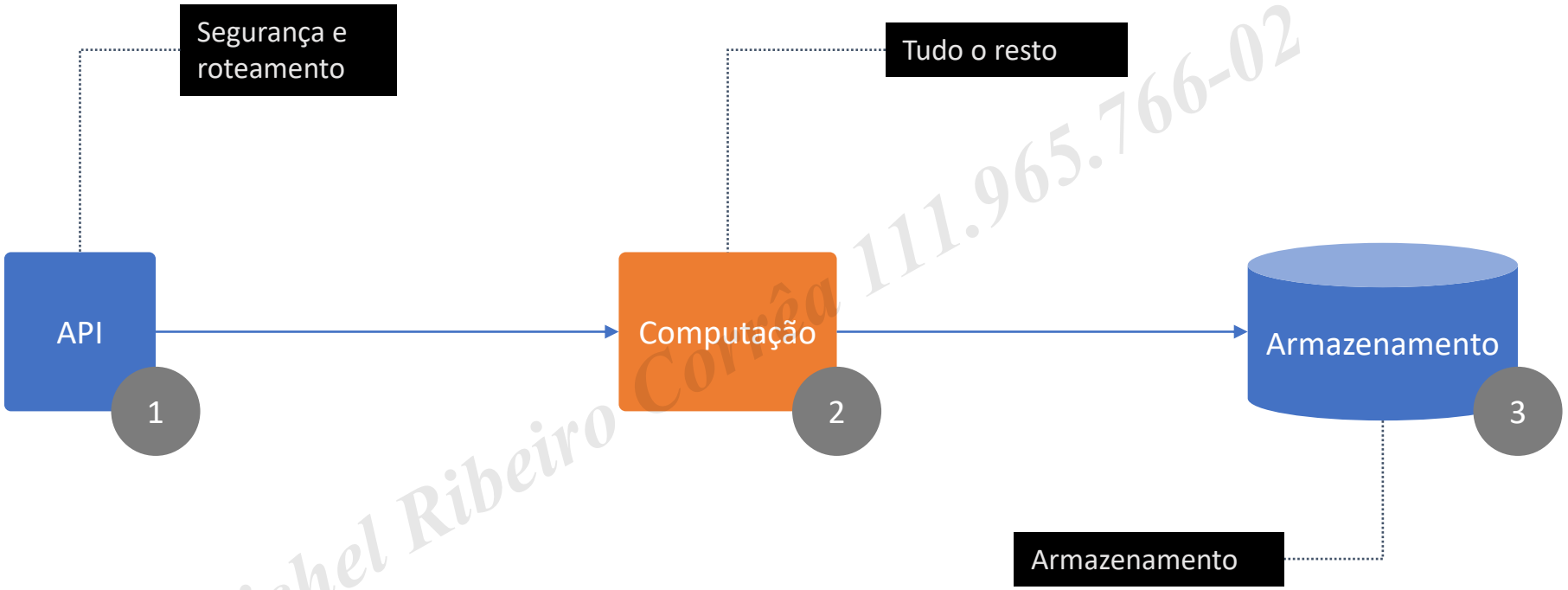
Elementos do aplicativo



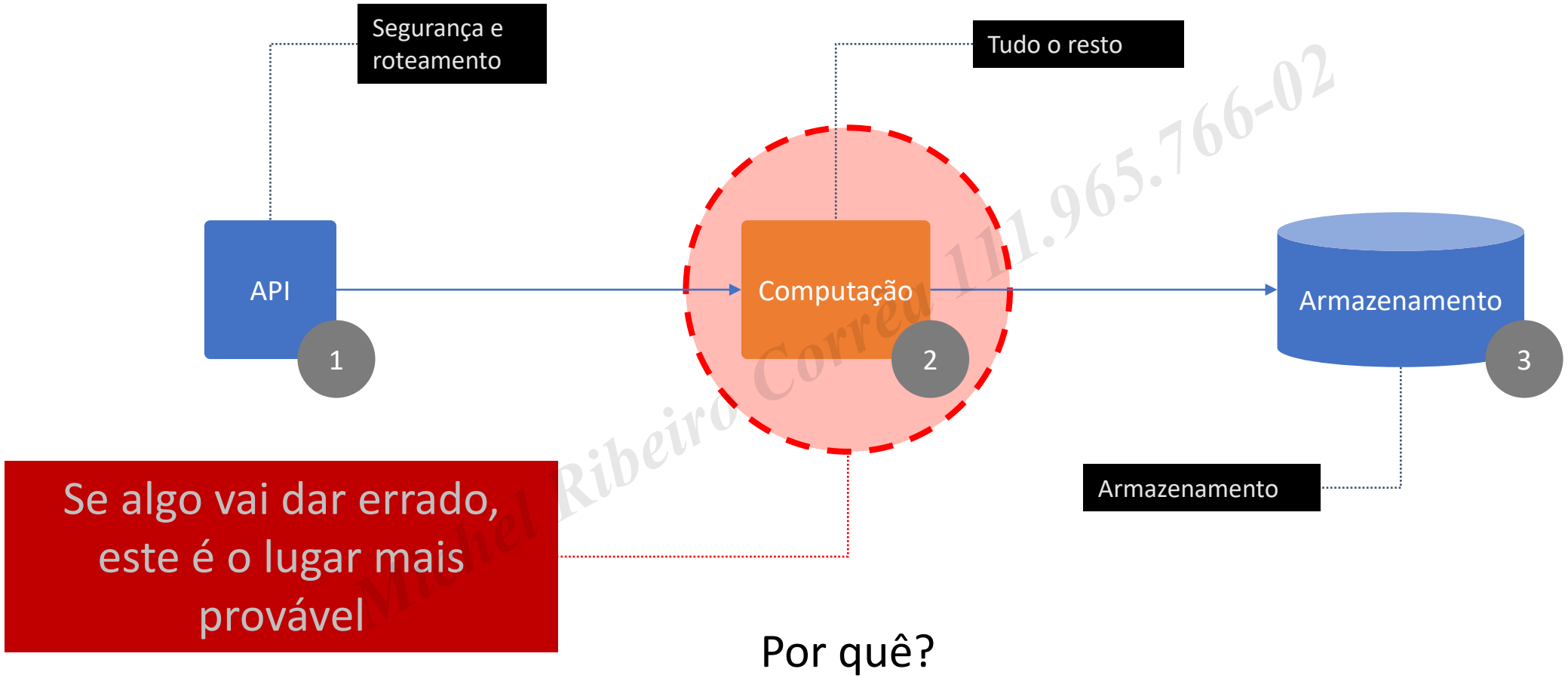
Elementos do aplicativo



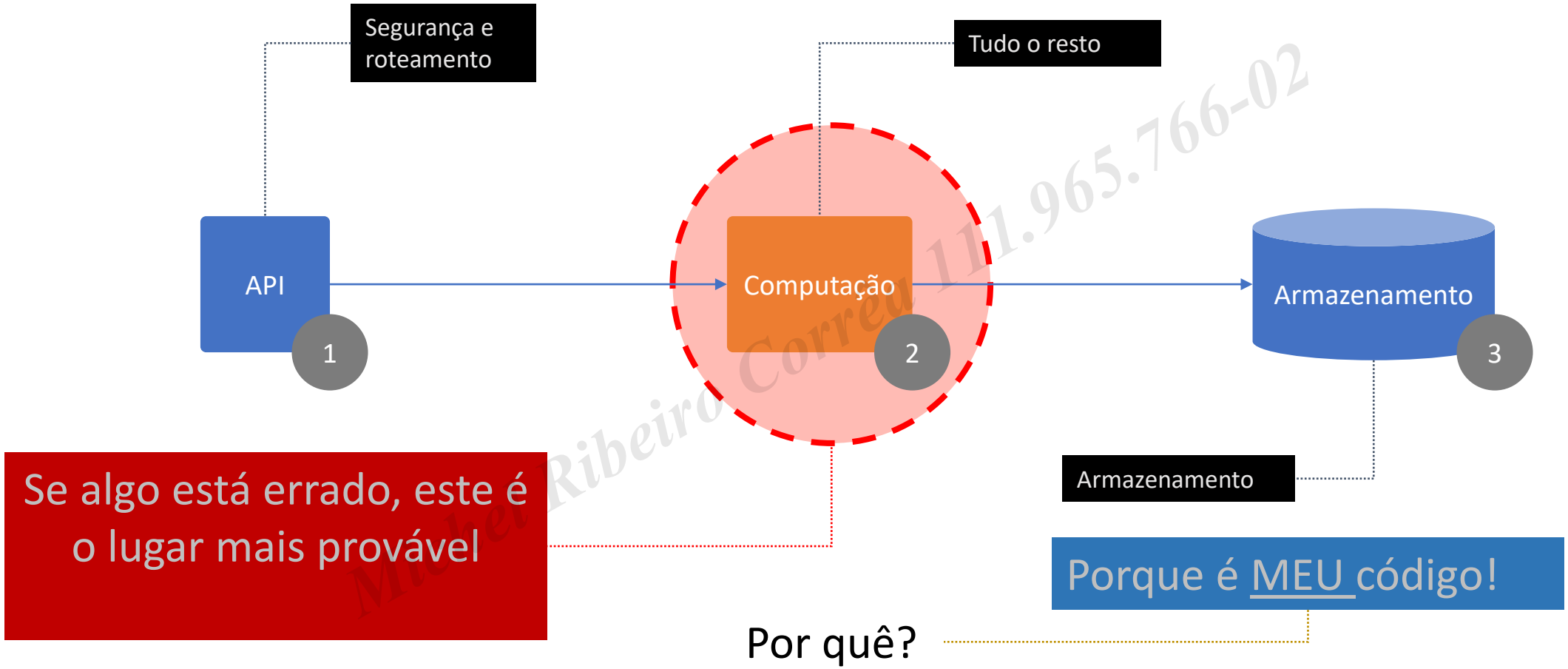
Elementos do aplicativo



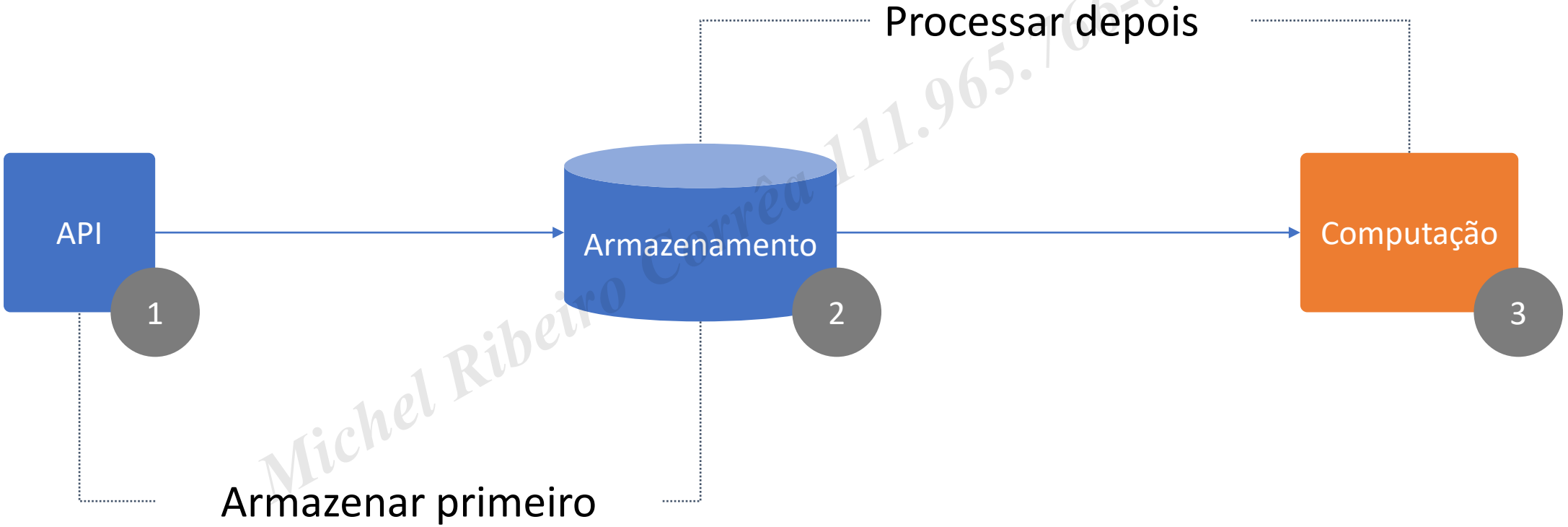
Elementos do aplicativo



Elementos do aplicativo

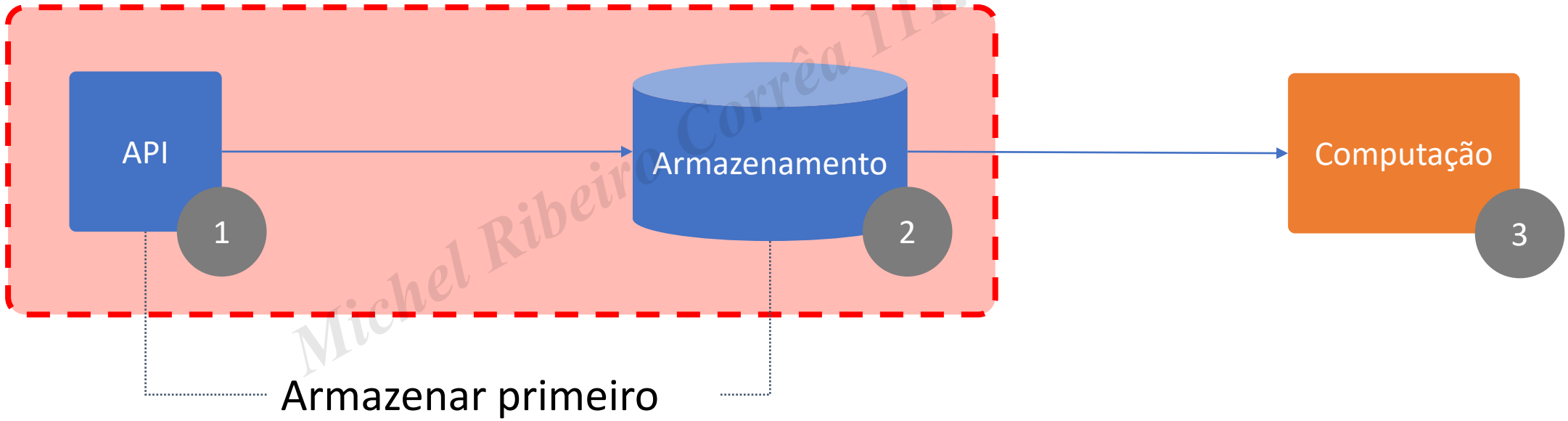


Pensando de forma assíncrona



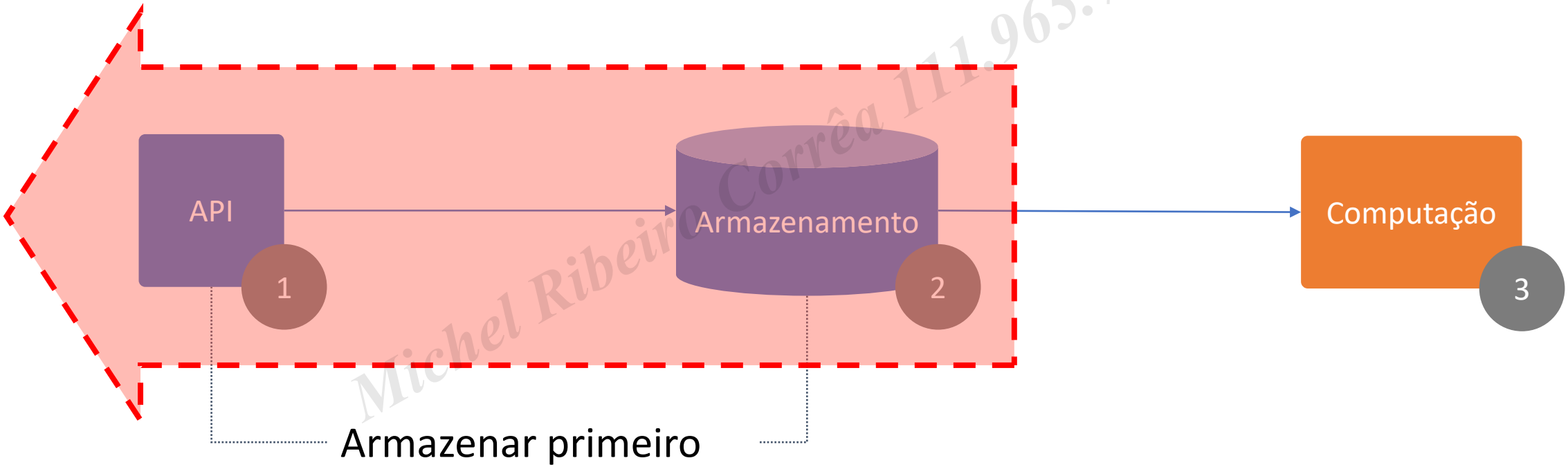
Pensando de forma assíncrona

- **Maior confiabilidade**: Os dados são armazenados antes que meu código os receba
- **Menos código**: Reduza o código por meio de integrações diretas de serviços



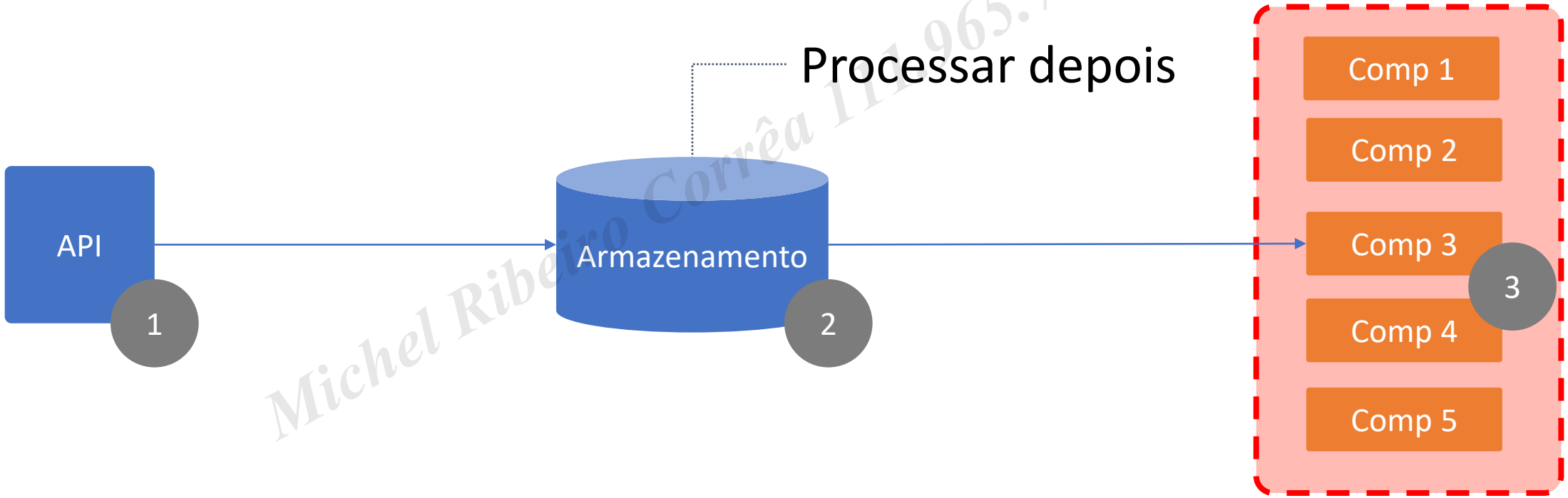
Pensando de forma assíncrona

- **Tempo de resposta mais rápido**: O cliente recebe uma confirmação e pode pesquisar mais dados, se necessário



Pensando de forma assíncrona

- **Faça mais**: O processamento assíncrono facilita fazer mais em menos tempo



Casos de Uso Orientados por Eventos (Event-Driven)

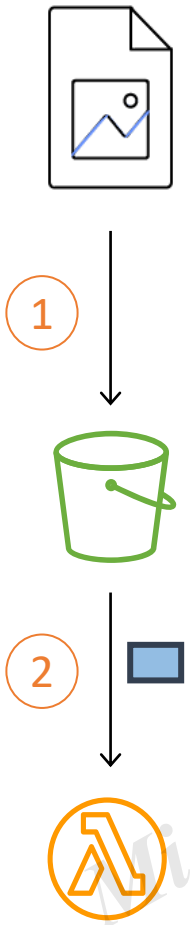
Processamento Assíncrono

PARTE 6

Processando uploads de arquivos

Exemplos: redimensionar fotos, extrair texto, traduzir, etc.

1. Objeto carregado para o Amazon S3 Bucket
2. Invocação **assíncrona** da função Lambda, a carga útil do evento inclui:
   - Nome do bucket
   - Chave de objeto

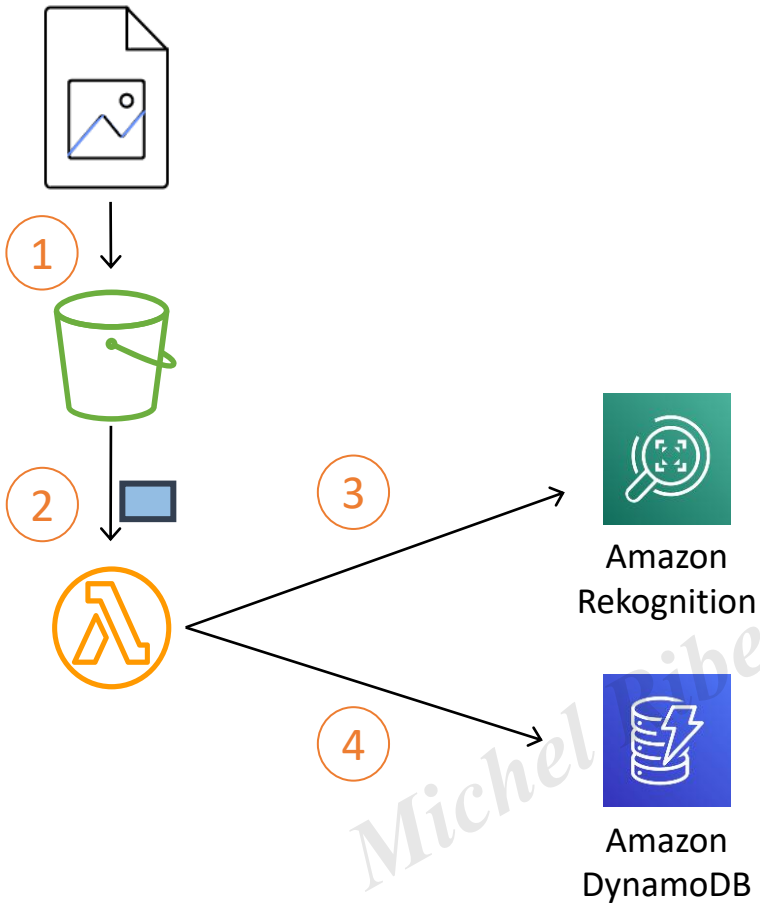


Processando uploads de arquivos

Exemplos: redimensionar fotos, extrair texto, traduzir, etc.

1. Objeto carregado para o Amazon S3 Bucket
2. Invocação **assíncrona** da função Lambda, a carga útil do evento inclui:
   - Nome do bucket
   - Chave de objeto

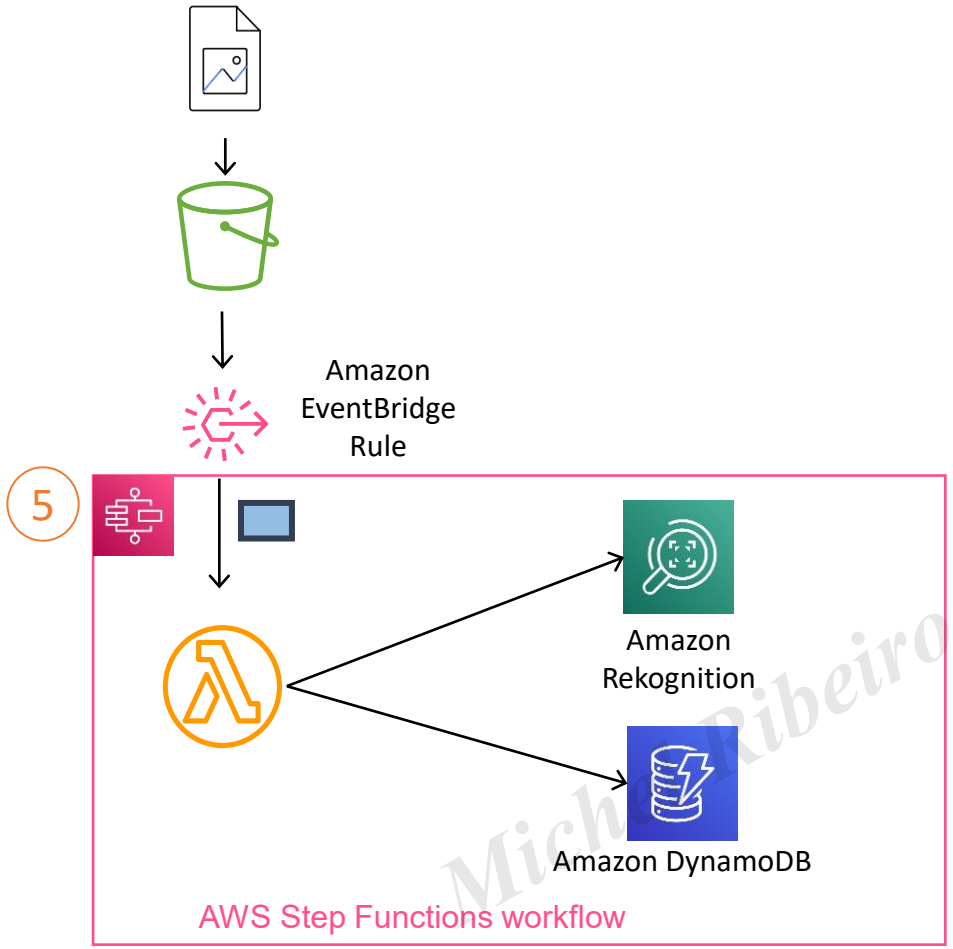
Processando uploads de arquivos, adicione rapidamente novas funcionalidades



Exemplo: Adicione análise de imagens e armazenamento de metadados

3. Analise fotos com a Amazon Rekognition

4. Armazene detalhes da imagem e resultados da análise

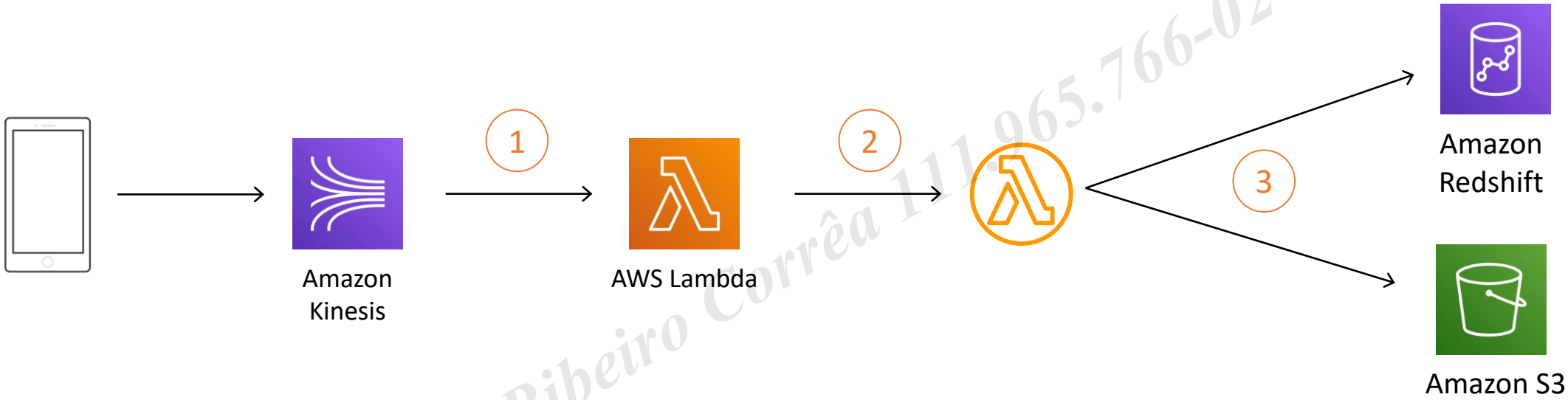


Processando uploads de arquivos, adicione rapidamente novas funcionalidades

Exemplo: Adicione análise de imagens e armazenamento de metadados

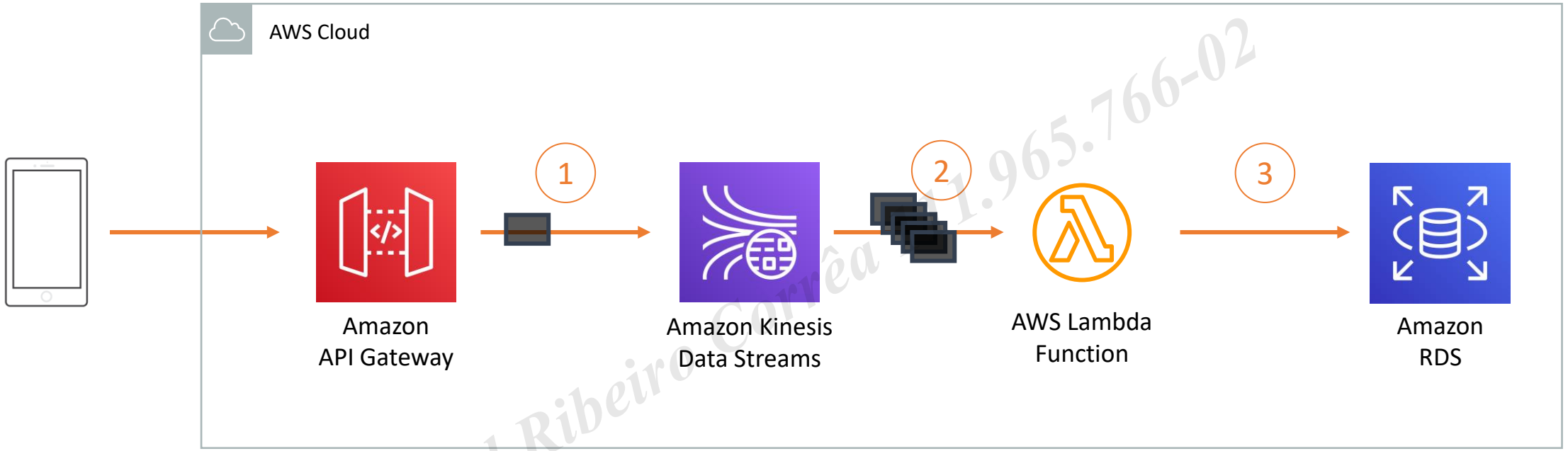
5. Simplifique a lógica de negócios com o Step Functions

Ingestão e armazenamento de dados de streaming



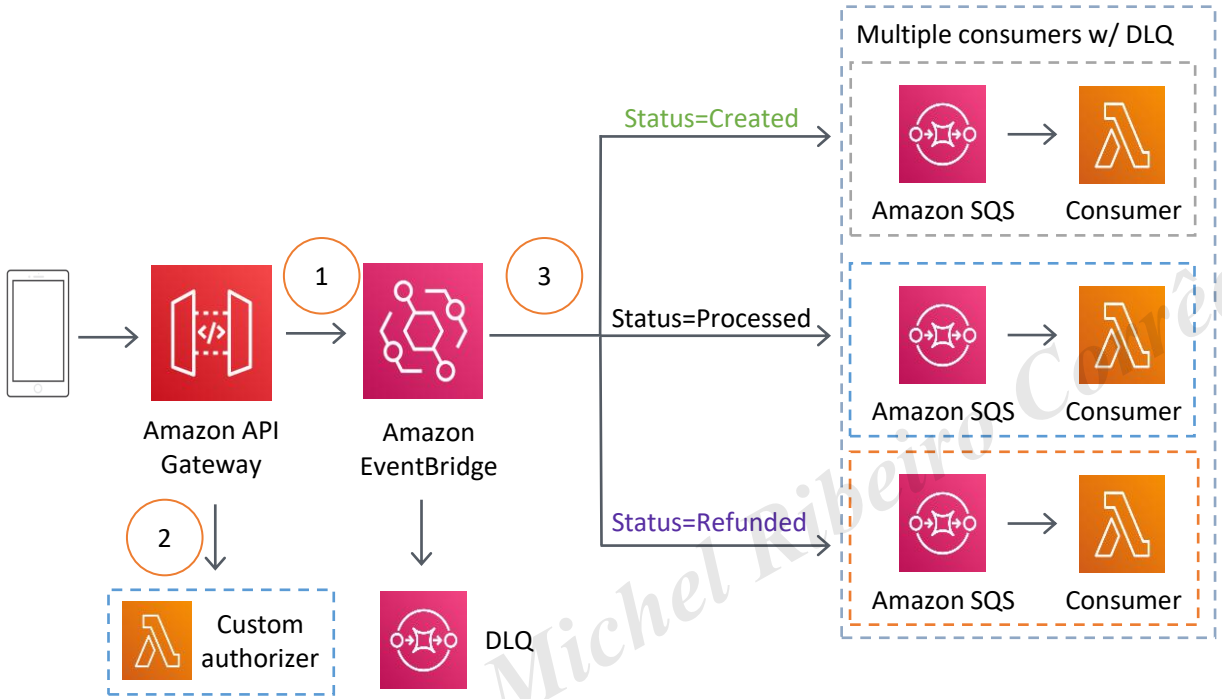
1. O serviço Lambda pesquisa mensagens do Kinesis Data Stream
2. A função é invocada de forma **síncrona** com **lotes** de mensagens
3. A função processa e/ou envia dados para armazenamentos de dados posteriores

Webhook “semi-serverless”



1. Padrão “armazenamento em primeiro lugar”: o API Gateway grava diretamente no Kinesis Data Stream
2. Kinesis como um buffer para limitar a concorrência downstream
3. Executar transação(ões) em lote

Fan-Out

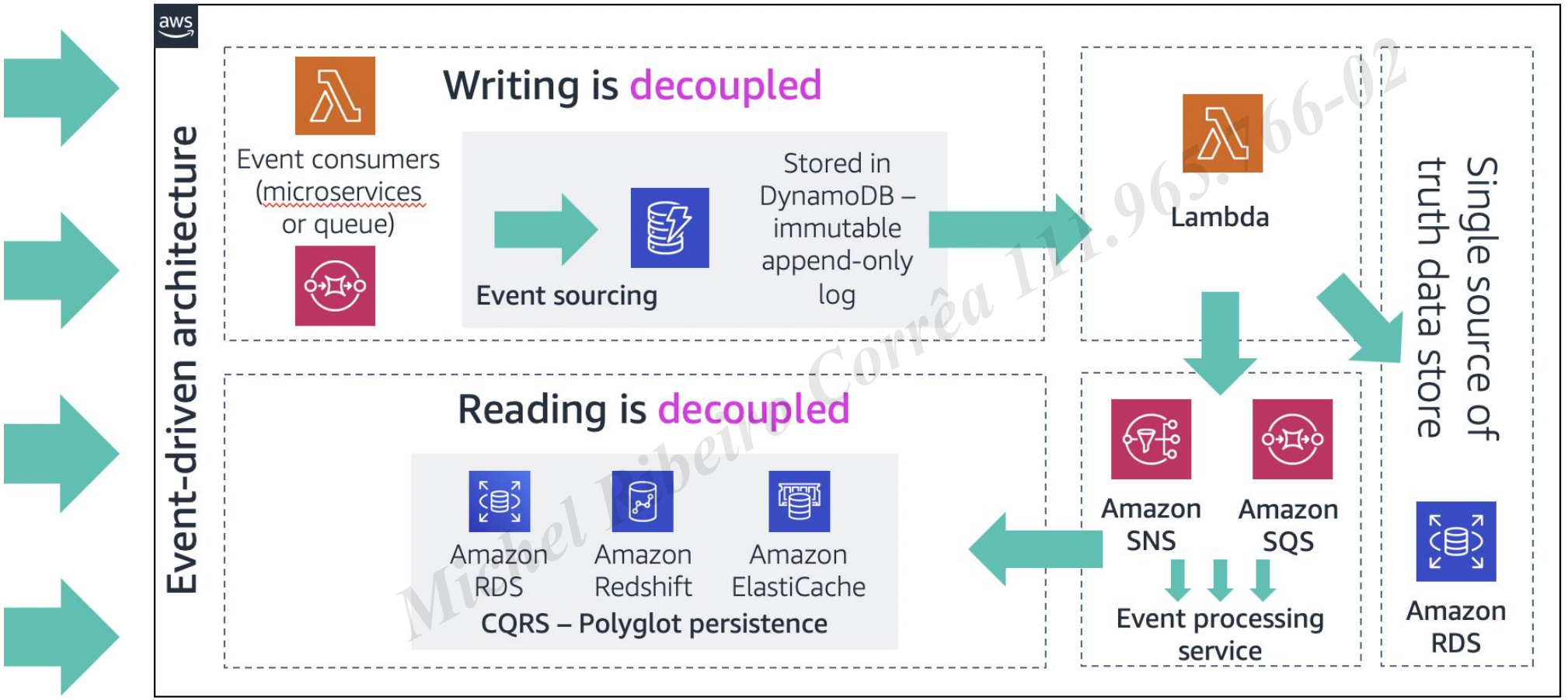


Fan-out distribui um evento para múltiplos consumidores e/ou envia atualizações para vários assinantes

1. "Armazenamento em primeiro lugar": integre o API Gateway diretamente ao EventBridge
2. Impor autorização
3. Use o roteamento para um processamento eficiente

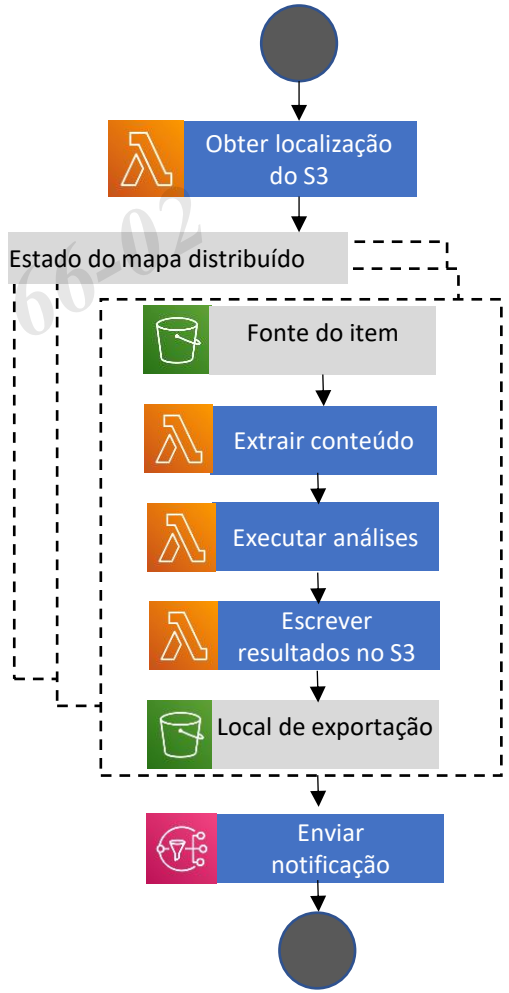
Também existe o padrão fan-in, que agrega múltiplos eventos

CQRS: Command Query Reponsibility Segregation



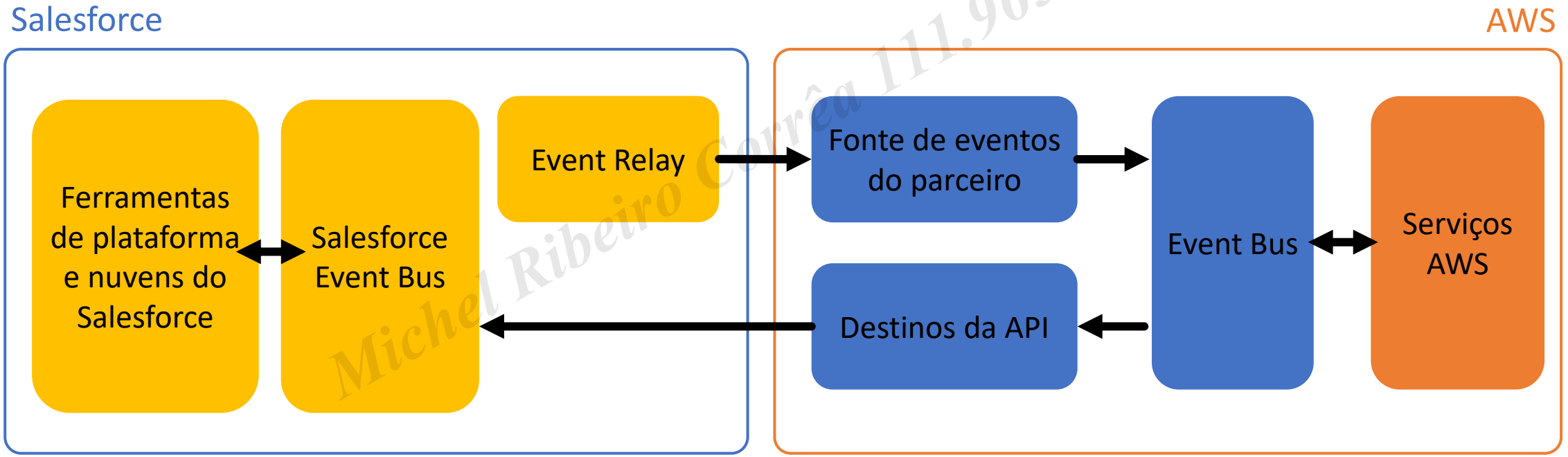
Processamento de dados em grande escala com o AWS Step Functions

- Processe dinamicamente grandes matrizes de dados com estado de mapa distribuído
- Coordene cargas de trabalho paralelas em larga escala no Step Functions
- Itere sobre milhões de objetos S3 como arquivos JSON ou CSV
- Até 10.000 execuções paralelas
- Invoque Lambda ou ECS/Fargate para computação serverless em larga escala sob demanda



Integre aplicativos de terceiros com a AWS

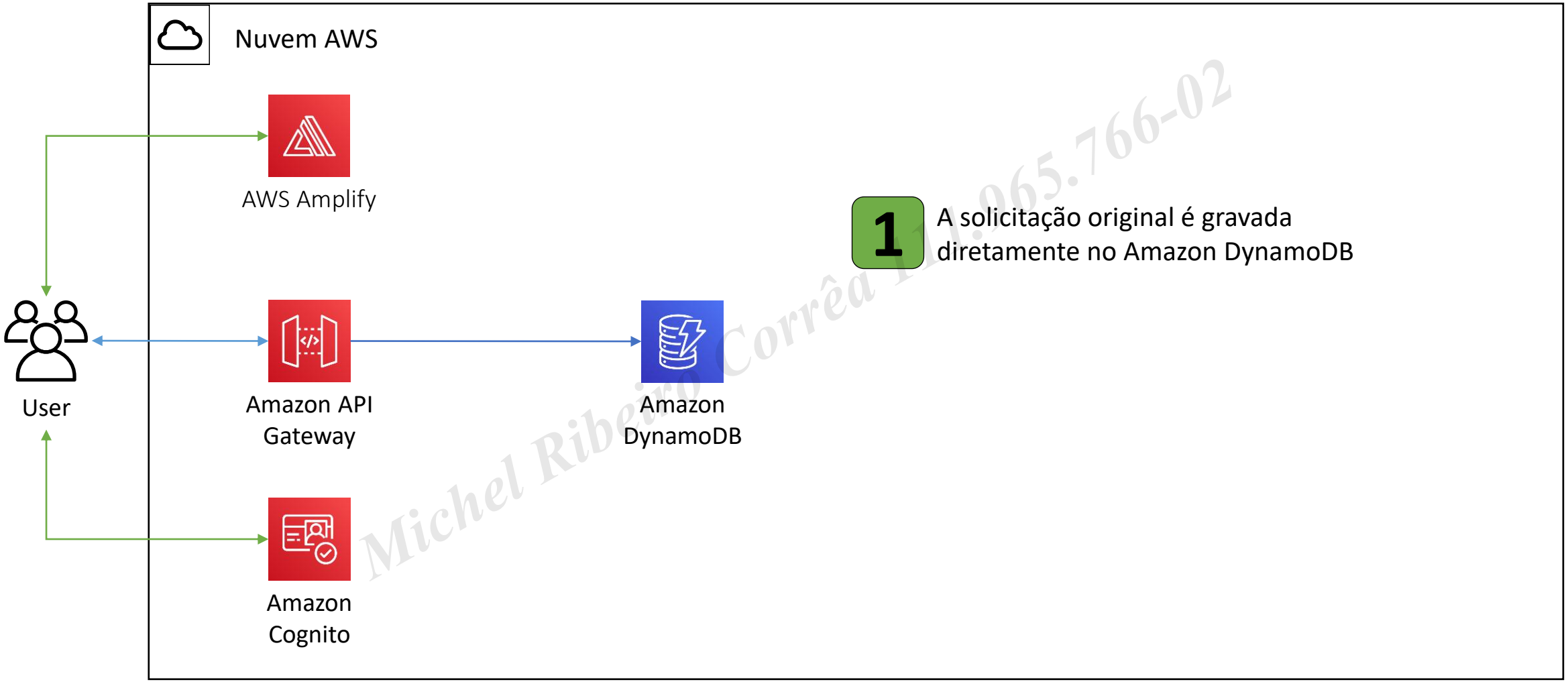
- Exemplo: integrar a Salesforce com a AWS
- Projetado para arquitetura orientada a eventos e aplicativos em tempo real.



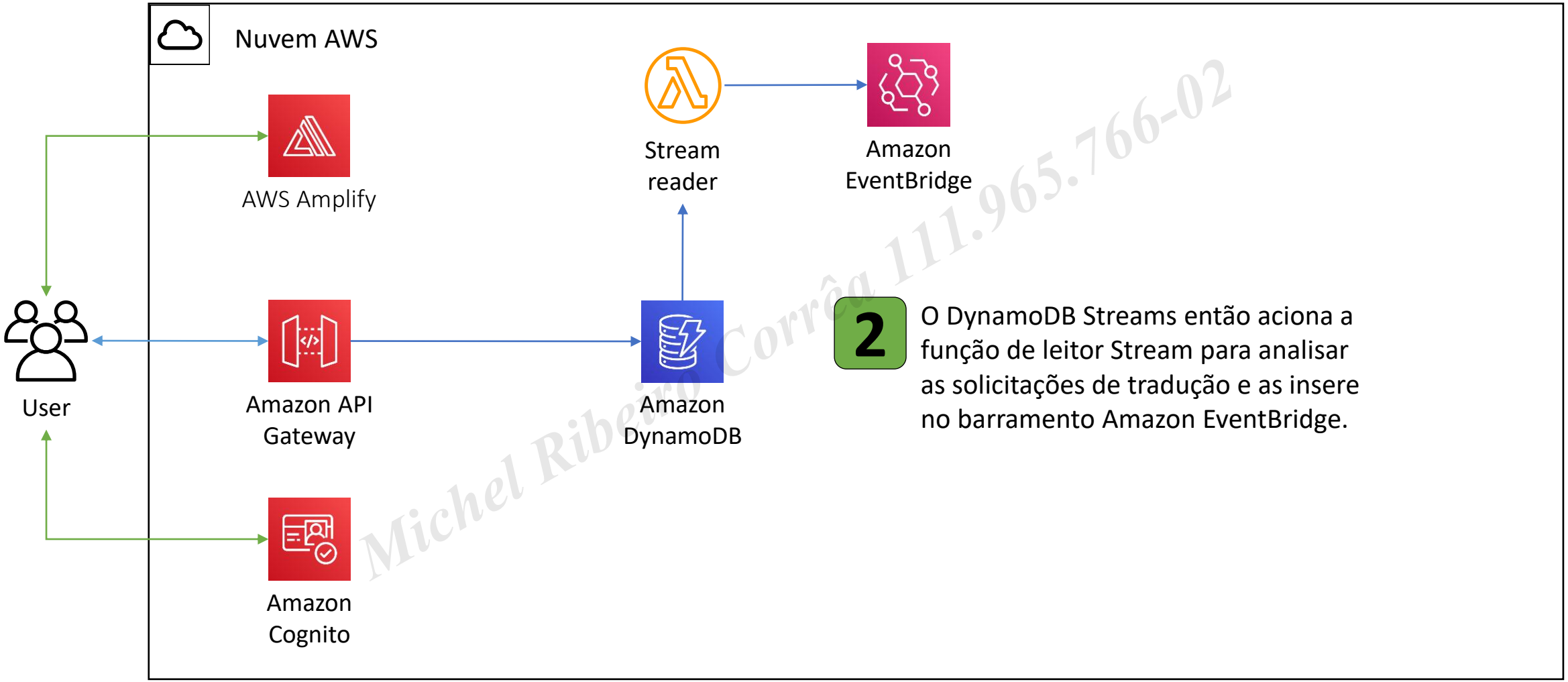
Aplicações assíncronas

PARTE 7

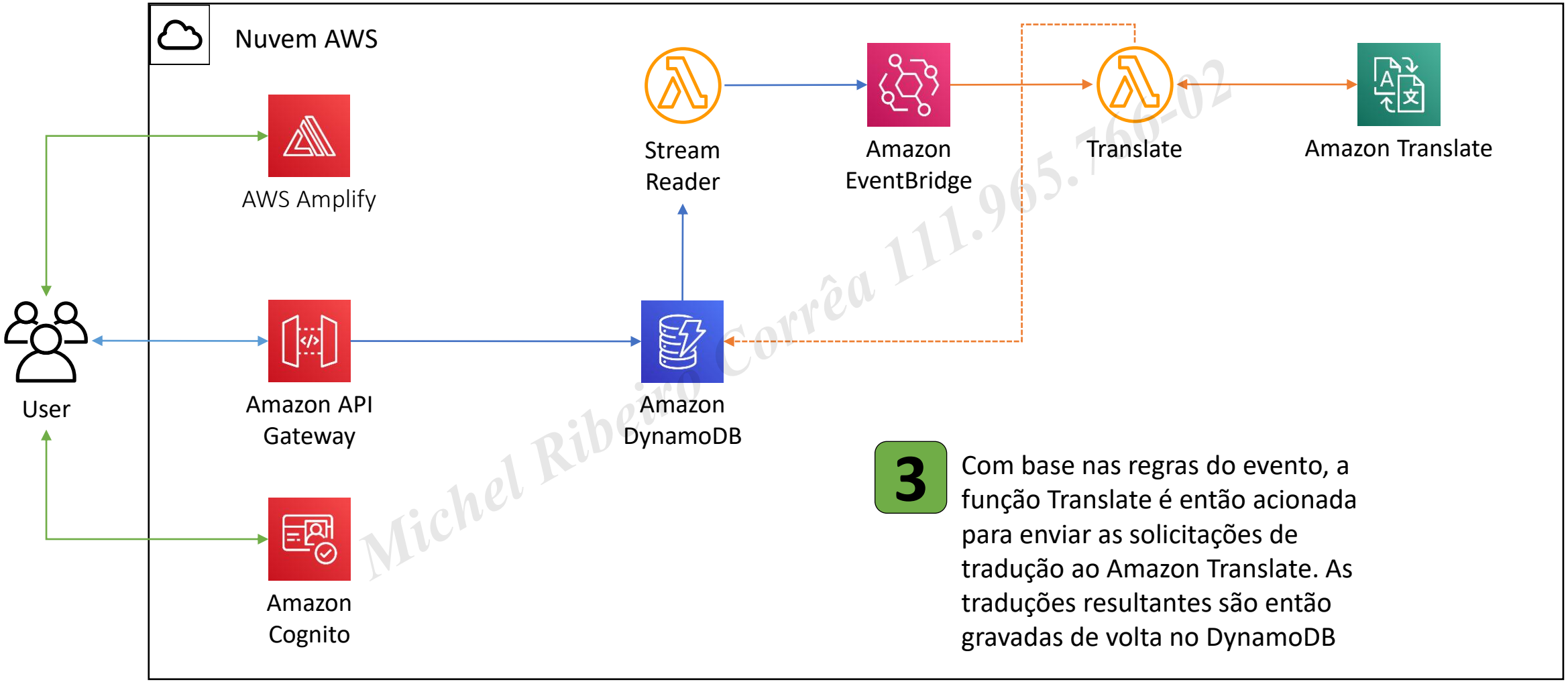
Exemplo de aplicativo assíncrono



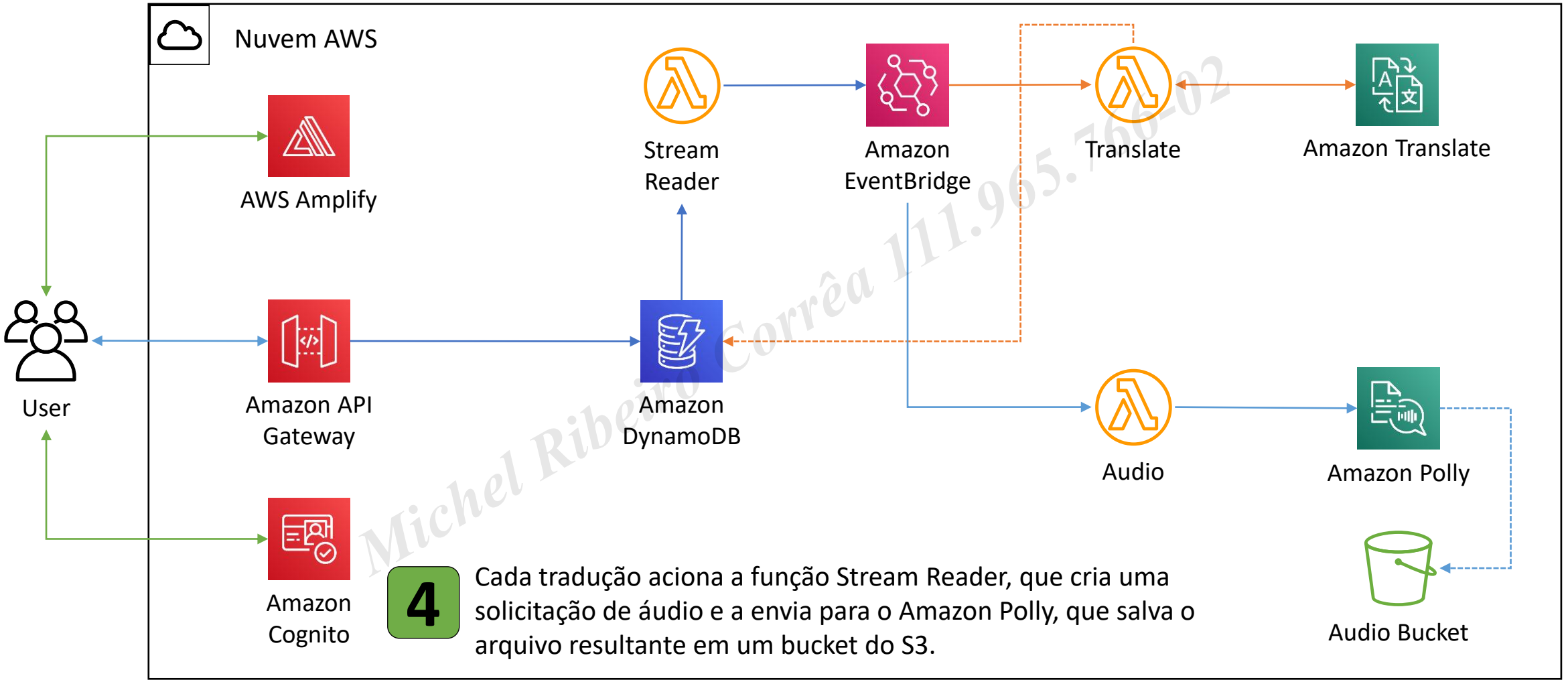
Exemplo de aplicativo assíncrono



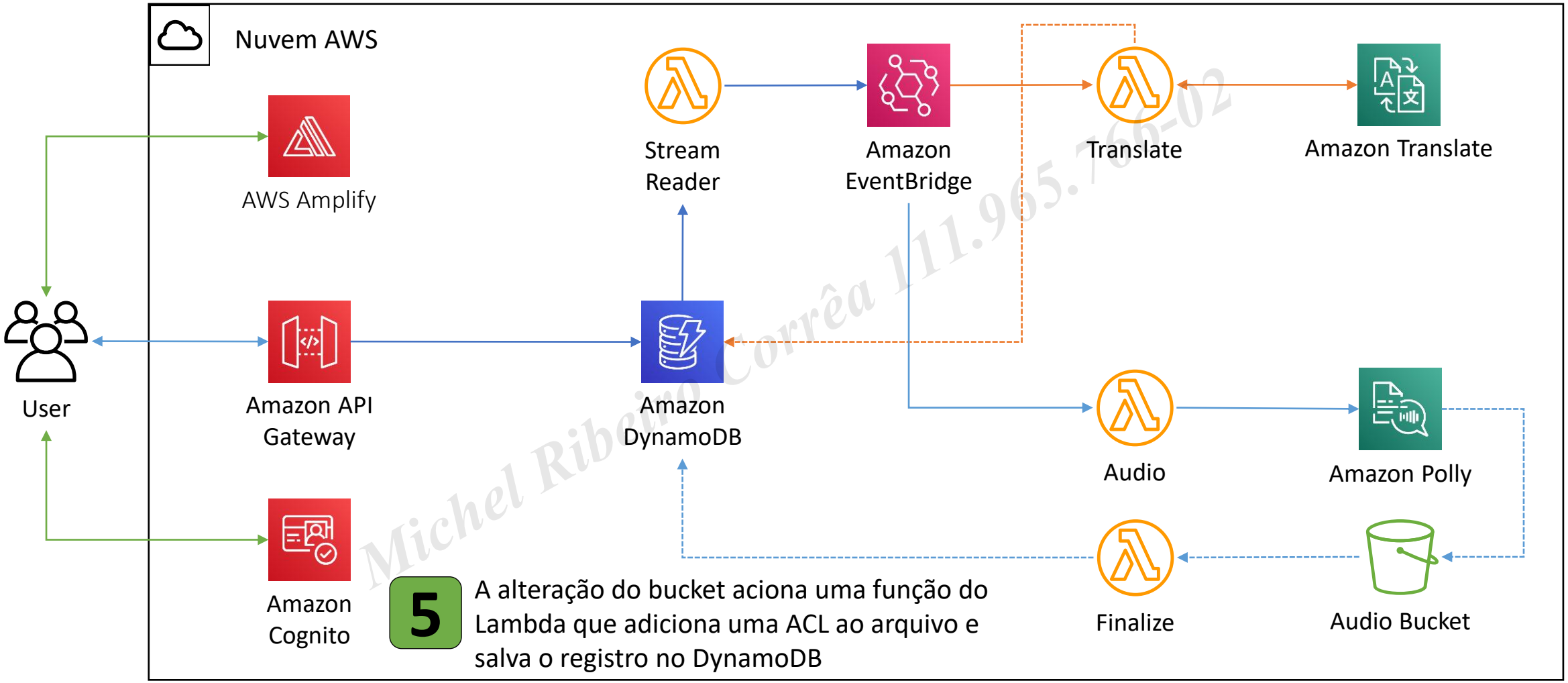
Exemplo de aplicativo assíncrono



Exemplo de aplicativo assíncrono

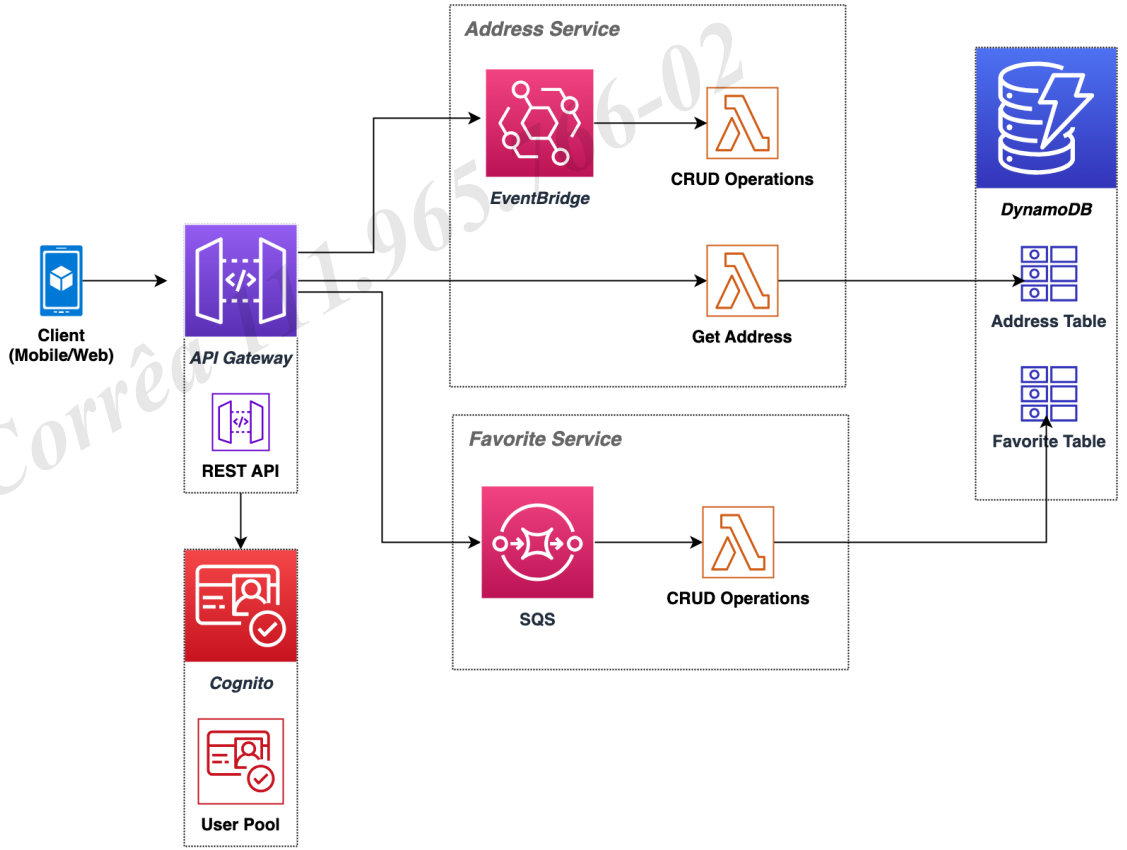


Exemplo de aplicativo assíncrono



Exercício

- Exemplo de aplicação assíncrona
- https://catalog.workshops.aws/serverless-patterns/en-US/module4



Encerramento

PARTE 8

Sugestões de Leituras/Referências

- (Introdução) Serverless Architectures with AWS Lambda - AWS Whitepaper: https://docs.aws.amazon.com/pt_br/whitepapers/latest/serverless-architectures-lambda/welcome.html
- AWS Serverless Youtube Channel: https://www.youtube.com/playlist?list=PL_N3d8yp5h5JvSdCkJ7EoKAhqM16Zyahi
- Serveless Land - (Este site reúne todos os blogs, vídeos e treinamentos mais recentes para AWS Serverless): https://serverlessland.com/
- AWS Lambda: https://aws.amazon.com/pt/lambda/
- New for AWS Lambda – 1ms Billing Granularity Adds Cost Savings: https://aws.amazon.com/pt/blogs/aws/new-for-aws-lambda-1ms-billing-granularity-adds-cost-savings/
- Getting Started with AWS Lambda Layers: https://dev.to/vealkind/getting-started-with-aws-lambda-layers-4ipk
- Optimizing AWS Lambda cost and performance using AWS Compute Optimizer: https://aws.amazon.com/pt/blogs/compute/optimizing-aws-lambda-cost-and-performance-using-aws-compute-optimizer/
- What is the best event source for doing pub-sub with AWS Lambda?: https://theburningmonk.com/2018/04/what-is-the-best-event-source-for-doing-pub-sub-with-aws-lambda/
- Serverless Microservice Patterns for AWS: https://www.jeremydaly.com/serverless-microservice-patterns-for-aws/

Sugestões de Leituras/Referências

- Operating Lambda: Application design and Service Quotas – Part 1: https://aws.amazon.com/pt/blogs/compute/operating-lambda-application-design-and-service-quotas-part-1/
- Operating Lambda: Application design – Scaling and concurrency: Part 2: https://aws.amazon.com/pt/blogs/compute/operating-lambda-application-design-scaling-and-concurrency-part-2/
- Operating Lambda: Application design – Part 3: https://aws.amazon.com/pt/blogs/compute/operating-lambda-application-design-part-3/
- Operating Lambda: Understanding event-driven architecture – Part 1: https://aws.amazon.com/pt/blogs/compute/operating-lambda-understanding-event-driven-architecture-part-1/
- Operating Lambda: Design principles in event-driven architectures – Part 2: https://aws.amazon.com/pt/blogs/compute/operating-lambda-design-principles-in-event-driven-architectures-part-2/
- Operating Lambda: Anti-patterns in event-driven architectures – Part 3: https://aws.amazon.com/pt/blogs/compute/operating-lambda-anti-patterns-in-event-driven-architectures-part-3/
- Operating Lambda: Building a solid security foundation – Part 1: https://aws.amazon.com/pt/blogs/compute/operating-lambda-building-a-solid-security-foundation-part-1/
- Operating Lambda: Building a solid security foundation – Part 2: https://aws.amazon.com/pt/blogs/compute/operating-lambda-building-a-solid-security-foundation-part-2/
- Choosing the right event-routing service for serverless: EventBridge, SNS, or SQS: https://lumigo.io/blog/choosing-the-right-event-routing-on-aws-eventbridge-sns-or-sqs/

Sugestões de Leituras/Referências

- Choosing between messaging services for serverless applications: https://aws.amazon.com/blogs/compute/choosing-between-messaging-services-for-serverless-applications/

Sessões:

- AWS re:Invent 2018: [REPEAT 1] A Serverless Journey: AWS Lambda Under the Hood (SRV409-R1): https://www.youtube.com/watch?v=QdzV04T_kec
- AWS re:Invent 2019: [REPEAT 1] Best practices for AWS Lambda and Java (SVS403-R1): https://www.youtube.com/watch?v=ddg1u5HLwg8
- AWS re:Invent 2019: [REPEAT 3] Serverless architectural patterns and best practices (ARC307-R3): https://www.youtube.com/watch?v=9IYpGTS7Jy0&t=2s
- (Session) - Optimizing Lambda Performance for Your Serverless Applications - AWS Online Tech Talks: https://www.youtube.com/watch?v=FTCaOQJvG6Y

Sugestões de Leituras/Referências

- Amazon SQS:
  https://aws.amazon.com/pt/sqs/
- AWS Lambda Adds Amazon Simple Queue Service to Supported Event Sources:
  https://aws.amazon.com/pt/blogs/aws/aws-lambda-adds-amazon-simple-queue-service-to-supported-event-sources/
- Amazon SQS as an Event Source to AWS Lambda: A Deep Dive:
  https://dzone.com/articles/amazon-sqs-as-an-event-source-to-aws-lambda-a-deep
- Amazon SNS:
  https://aws.amazon.com/pt/sns/
- Introducing message archiving and analytics for Amazon SNS:
  https://aws.amazon.com/pt/blogs/compute/introducing-message-archiving-and-analytics-for-amazon-sns/?nc1=b_rp
- Working with delivery stream destinations:
  https://docs.aws.amazon.com/sns/latest/dg/firehose-working-with-destinations.html
- Introducing Amazon SNS FIFO – First-In-First-Out Pub/Sub Messaging:
  https://docs.aws.amazon.com/sns/latest/dg/firehose-working-with-destinations.html
- Building event-driven architectures with Amazon SNS FIFO:
  https://aws.amazon.com/pt/blogs/compute/building-event-driven-architectures-with-amazon-sns-fifo/
- Migrating message driven applications to Amazon MQ for RabbitMQ:
  https://aws.amazon.com/pt/blogs/compute/migrating-message-driven-applications-to-amazon-mq-for-rabbitmq/

Sugestões de Leituras/Referências

AWS Serverless Application Repository

- Building serverless apps with components from the AWS Serverless Application Repository: https://aws.amazon.com/pt/blogs/aws/building-serverless-apps-with-components-from-the-aws-serverless-application-repository/

Obrigado!

Prof. José Maria Cesário Jr. | linkedin.com/in/josemariacesariojunior/

MBA USP ESALQ