Uma imagem contendo texto

Descrição gerada automaticamente

Mainframe:

Sistema legado que fornece os dados para a avaliação.

Listener:

Para que não seja preciso modificar o sistema legado, o “Listener” seria responsável por ouvir o barramento do “Mainframe” e efetuar a coleta dos dados gerados por ele.

Informações MF:

Utilizando **Amazon Kinesis Data Firehose**, tem o papel de transmitir as informações coletadas pelo “Listener” para o “Controller”.

Controller:

Consulta no “BD 1” as regras equivalentes a transação que foi coletada e envia os dados, tanto da transação como das regras, para a “Central de risco”. Assim que recebe o resultado da avaliação feita pela “Central de Risco” envia para “Resultado das avaliações”. Utilizaria o **AWS Lambda** para administrar esses fluxos.

BD 1:

Base de dados **Amazon Simple Store Service** contendo as regras de negócio definidas pela área responsável.

Portal de regras:

Portal front-end feito em **Angular** disponibilizado para melhor gerenciamento das regras pela área responsável. Dessa forma mantém a inclusão, exclusão e edição dos dados do “BD 1” o mais “user friendly” possível.

Central de risco:

Consulta o “BD 2”, que contém o histórico das avaliações e fórmulas para efetuar as avaliações, e efetua a avaliação necessária. Utilizando o **Amazon SageMaker** há a possibilidade de criar, treinar e implantar um modelo de machine learning que melhore o processo de avaliação para reduzir os casos de dúvida que precisam de avaliação humana.

BD 2:

Base de dados **Amazon Simple Store Service** com as fórmulas para efetuar a avaliação e o histórico de avaliações passadas.

Resultado das avaliações:

Recebe as avaliações que foram efetuadas e por meio do **Amazon Kinesis Data Firehose** e envia para o “BD 3”.

BD 3:

Terá o resultado das avaliações que foram feitas pela plataforma. Se houver dúvidas sobre determinada avaliação que precise de validação humana, ela será feita pelo “Portal de backoffice” que irá atualizar os dados diretamente “BD 3”.

Também utilizaria o **Amazon Simple Store Service**.

Portal para backoffice:

Portal front-end em **Angular** responsável por apresentar as avaliações feitas, possibilitar a revisão das avaliações que apresentaram dúvida e dar a opção de gerar relatórios para a área de negócio responsável ou qualquer outra que faça uso dessas informações.

Pós avaliação:

Tem o papel de incrementar dados no “BD 2” com as novas avaliações que foram feitas e atualizar as avaliações que apresentaram dúvidas após análise humana para que a central de risco identifique esses casos e melhore seu modelo baseado na ferramenta de machine learning implementada. Para isso podemos utilizar o **Amazon Kinesis Data Firehose**.

Data Science:

Responsável por avaliar a massa de dados gerados pós avaliação e utilizando o **Amazon QuickSight** apresentar insights referente as transações que foram feitas.

**Perguntas**

1. Qual é a sua estratégia para modernizar esse legado?

Apenas consumir os dados do mainframe sem inserir novas funcionalidades a ele e implementar uma nova arquitetura na nuvem (AWS) para efetuar as avaliações necessárias, utilizando novas tecnologias de banco de dados, machine learning, streaming de dados e etc.

1. Escolha uma funcionalidade prioritária para modernizar.

Foi escolhido o processo de avaliação de prevenção a fraudes.

1. Quais os critérios você adotou para priorizar a funcionalidade a ser modernizada?

- Se possível não alterar o legado ou mexer o mínimo possível.   
- Utilizar novas tecnologias que sejam fáceis de manter e de escalar.   
- Manter os processos isolados e de forma de fácil administração para as equipes responsáveis.   
- Uma arquitetura que consiga conviver com o legado por muito tempo até encontrar uma opção para a atualização do legado.

1. Quais seriam os requisitos imprescindíveis para essa funcionalidade?

- Confiabilidade que o sistema cumpra seu papel mesmo em casos extremos. - Segurança para que as informações sobre as transações não sejam vazadas.

1. Como será a convivência desta funcionalidade modernizada com o legado? (Visão técnica)

Para o legado nada muda já que iremos apenas consumir seu barramento para coletar os dados produzidos por ele. O modelo foi pensado em manter funções bem definidas sem alterar o legado para que no momento que o legado for substituído por algo mais atualizado isso não cause o mínimo impacto ao restante da aplicação que já vai estar funcionando.

1. Quais serão os fatores críticos para o sucesso da modernização dessa funcionalidade?  
   - Garantir que mesmo com a modernização da funcionalidade o sistema continue funcionando da maneira esperada.   
   - Garantir que o novo modelo facilite a vida dos usuários, diminuindo as validações humanas, deixando um modelo “user friendly” pelos portais das áreas de negócio e diminuir ao máximo as dificuldades que eram causadas pelo modelo antigo. Não podemos entregar algo novo e bom no ponto de vista tecnológico, mas que para os usuários seja pior.  
   - Os dados gerados devem ser estudados para que, além de possibilitar maior controle e visão sobre as avaliações, possuem grande valor de mercado para novos produtos ou estratégias de mercado.
2. Quais seriam as linguagem e tecnologias envolvidas na solução e por quê?

Linguagens:

Java para a o back-end e integração com o front-end dos portais, para montar uma plataforma web. Será simples de montar um ambiente dessa natureza já que a própria linguagem fornece ferramental para facilidade de configurações e ambiente (exemplo: spring). Para o front-end utilizaria o Angular que é simples e fácil de ser utilizado junto com o java.

Python para montar as fórmulas necessárias pois há muitos algoritmos implementados nativamente da biblioteca do python com foco de análise de dados e estatística.   
Tecnologias:

AWS para que toda a arquitetura de avaliação de fraudes fique na nuvem. Dessa forma conseguimos focar no desenvolvimento e deixar outros fatores como escalabilidade, compra de máquinas, rede, disponibilidade, por conta do fornecedor do serviço.

Amazon Kinesis Data Firehose para fazer o streaming e avaliação de dados de maneira rápida e segura.

Amazona Lambda para executarmos o código sem a necessidade de gerenciar um servidor.

Amazon Simple Storage Service para armazenar objetos com alta performance, segurança e disponibilidade.

Amazon SageMaker para facilitar a criação, treino e implantação de um modelo de machine learning e melhorarmos o modelo de avaliação, tornando-o mais assertivo e com o mínimo de validação humana.

Amazon QuickSight efetuar avaliações sobre os dados gerados pelas avaliações e apresentar um insight com base nessas informações para que sejam possíveis diversas ações de melhoria e estratégias de mercado com base no que for apresentado.

1. Como você irá armazenar esses dados (tecnologias e modelos)?

Como tecnologia utilizaria o Amazon Simple Store Service que efetua um armazenamento de objetos por bucket. Dessa forma o modelo descrito abaixo para cada banco de dados se baseia nessa tecnologia.

No “BD 1”, banco de regras, cada cliente teria um bucket nomeado pelo Client ID que é um código único para identificar o cliente. Dessa forma, há a facilidade de encontrar as regras do cliente apenas por seu Client ID e deixamos as regras de maneira isolada, ou seja, se precisar alterar alguma regra certificamos que é de um cliente em específico sem mudar o comportamento dos demais.

No “BD 2”, banco com histórico das transações e modelos de avaliação, manteria cada cliente com seu bucket identificado por seu Client ID e armazenaria o histórico de cada um deles eu seu bucket. Teria outro bucket para armazenar apenas os modelos que são usados para avaliação, dessa forma acessamos os modelos de uma forma mais direta com consultas simples já que as respostas para a central de risco não podem demorar. Se a quantidade de modelos for muito grande e muito diferente umas das outras estudaria a possibilidade de dividir os modelos em mais buckets.

No “BD 3”, banco de registro das transações, cada cliente também teria seu bucket mas além do Client ID para identificar o bucket teríamos o Event ID específico da transação do cliente. Dessa forma, conseguimos organizar as transações de cada cliente pelo Event ID em seu bucket em específico.

1. Como a performance será otimizada de um jeito que ficará fácil de escalar?

Utilizando as tecnologias apresentadas na resposta da pergunta 7, o que realmente precisa de atenção é o código central, o core da aplicação, pois os serviços de AWS irão garantir boa performance e fácil escalabilidade, se necessária, dos serviços.

1. Dado que é um projeto importantíssimo para a organização, como você faria para acelerar o desenvolvimento, entregando valor para o cliente em menor tempo?

Apesar de poder apresentar um alto custo financeiro levar toda a plataforma para a nuvem, acredito que seja um bom passo para apresentar resultados rápidos e que o foco seja total no desenvolvimento e manutenção do código que trata do negócio. Pontos como rede, disponibilidade de máquina, deploys, planos de backup, são responsabilidade do fornecedor de serviço.

1. Criar uma solução simples no GitHub ou qualquer repositório open source voltado à microserviços que utilize a stack java, Quarkus ou Eclipse Vertx, podendo utilizar um banco como MongoDB. A solução não será buildada local porem faremos analise e questionaremos da solução em cima do código fonte durante a entrevista com o candidato.