Arquitectura de sistemas de banca por internet

Descripción del sistema

El sistema de banca por internet que se propone en el ejercicio tiene como objetivo permitir que los usuarios accedan a sus cuentas bancarias y realicen diversas operaciones de forma remota. El sistema se integrará con dos sistemas existentes: una plataforma Core que contiene información básica de clientes, movimientos y productos, y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle.

Componentes del sistema

El sistema de banca por internet estará compuesto por los siguientes componentes:

Capa de presentación: Esta capa estará compuesta por dos aplicaciones: una aplicación web de una sola página (SPA) y una aplicación móvil desarrollada en un framework multiplataforma. Ambas aplicaciones permitirán a los usuarios interactuar con el sistema y realizar las operaciones deseadas.

Capa de servicios: Esta capa estará compuesta por una serie de servicios web que proporcionarán la funcionalidad del sistema. Estos servicios estarán expuestos a través de interfaces en formato JSON o XML.

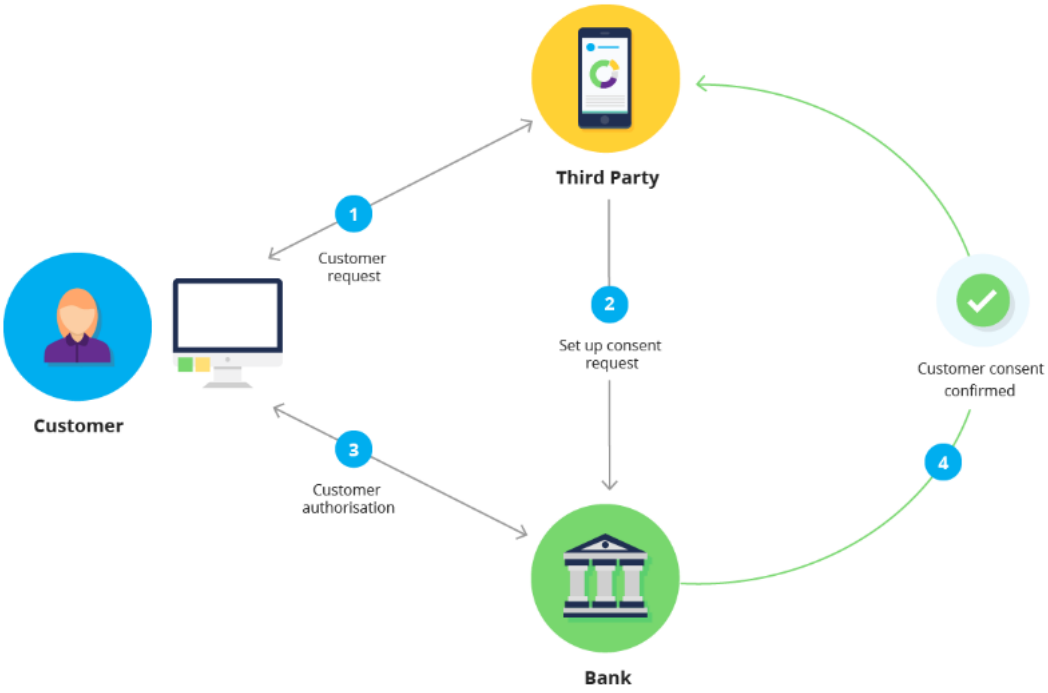
Capa de acceso a datos: Esta capa se encargará de acceder a los datos de los sistemas Core y del sistema independiente. Se utilizará un ORM (Object Relational Mapper) para mapear los objetos del sistema a las tablas de las bases de datos.

Capa de integración: Esta capa se encargará de integrar el sistema de banca por internet con los sistemas Core y el sistema independiente. Se utilizarán protocolos de mensajería como SOAP o REST para la comunicación entre los sistemas.

Capa de autenticación y autorización: Esta capa se encargará de autenticar a los usuarios y autorizarlos para realizar las operaciones deseadas. Se utilizará el estándar OAuth 2.0 para la autenticación y se integrará con el sistema de reconocimiento facial existente para el onboarding de nuevos clientes en la aplicación móvil.

Flujo de autenticación

El siguiente diagrama muestra el flujo de autenticación del sistema:



Authentication flow

El usuario ingresa su nombre de usuario y contraseña en la aplicación web o en la aplicación móvil.

La aplicación envía las credenciales del usuario al servicio de autenticación.

El servicio de autenticación valida las credenciales del usuario contra el sistema Core.

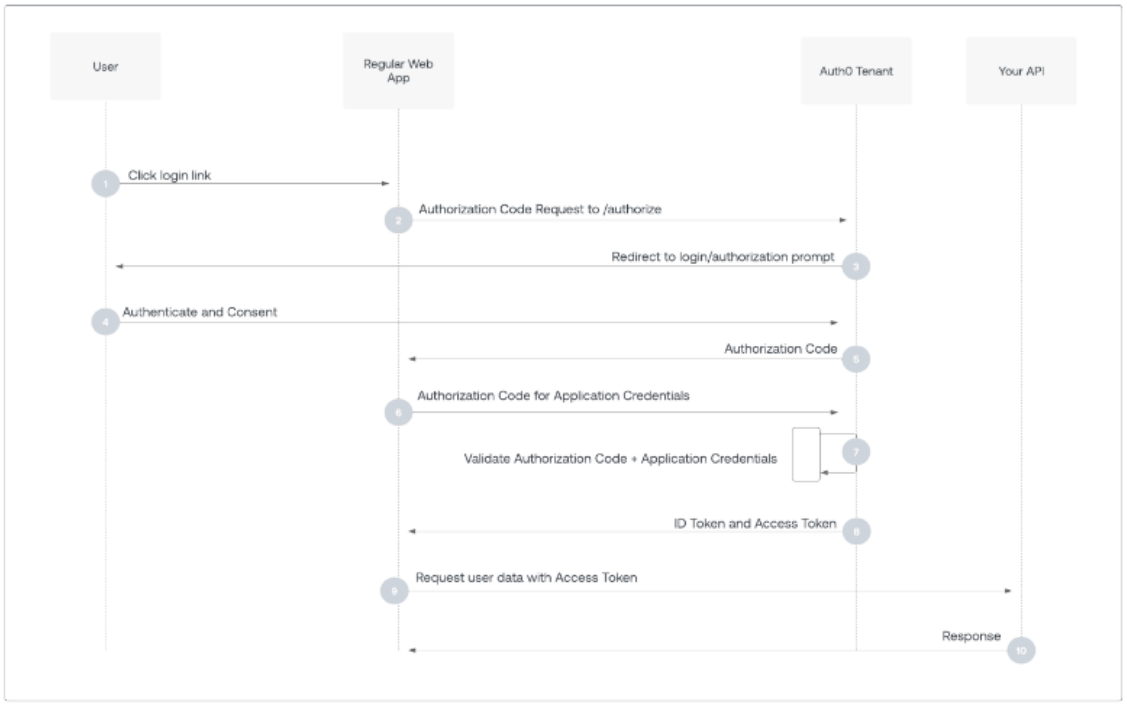
Si las credenciales son válidas, el servicio de autenticación genera un token de acceso y lo envía a la aplicación.

La aplicación almacena el token de acceso en el dispositivo del usuario.

La aplicación utiliza el token de acceso para realizar solicitudes a los servicios web del sistema.

Flujo de autorización

El siguiente diagrama muestra el flujo de autorización del sistema:



Authorization flow

La aplicación envía una solicitud a un servicio web junto con el token de acceso del usuario.

El servicio web valida el token de acceso y determina si el usuario tiene los permisos necesarios para realizar la operación solicitada.

Si el usuario tiene los permisos necesarios, el servicio web procesa la solicitud y devuelve una respuesta a la aplicación.

Si el usuario no tiene los permisos necesarios, el servicio web devuelve un error a la aplicación.

Consideraciones de seguridad

El sistema de banca por internet debe cumplir con los más altos estándares de seguridad para proteger la información de los usuarios. Algunas de las medidas de seguridad que se deben implementar son las siguientes:

Autenticación fuerte: Se debe utilizar un mecanismo de autenticación fuerte, como OAuth 2.0 con reconocimiento facial, para proteger las cuentas de los usuarios.

Encriptación: Toda la comunicación entre el sistema y los clientes debe estar encriptada para proteger contra ataques de intercepción.

Control de acceso: Se debe implementar un sistema de control de acceso basado en roles para permitir que los usuarios solo accedan a los recursos que necesitan.

Auditoría: Se deben registrar todas las actividades del sistema para poder detectar y prevenir actividades fraudulentas.

Arquitectura C4.

Diagrama de Contexto

El diagrama de contexto muestra todos los sistemas involucrados y sus interacciones a alto nivel:

- **User (Usuario Final)**:
  - **Descripción**: Representa al usuario que interactúa con el sistema desde un navegador web o una aplicación móvil.
  - **Función**: Inicia solicitudes de acceso al sistema, como iniciar sesión, consultar movimientos, realizar transferencias, etc.

- **Route53 (DNS y Balanceo de Carga)**:
  - **Descripción**: Amazon Route 53 es un servicio de DNS y balanceo de carga en la nube de AWS.
  - **Función**: Se encarga de dirigir el tráfico de los usuarios hacia los servicios adecuados mediante la resolución de nombres de dominio y el balanceo de carga.

- **CloudFront (CDN)**:
  - **Descripción**: Amazon CloudFront es una red de entrega de contenidos (CDN) de AWS.
  - **Función**: Distribuye el contenido de la aplicación (como archivos estáticos) de manera rápida y eficiente a nivel global, reduciendo la latencia.

- **ELB (Elastic Load Balancer)**:
  - **Descripción**: Un servicio de balanceo de carga que distribuye automáticamente el tráfico de entrada entre varias instancias de aplicación.
  - **Función**: Asegura que el tráfico se distribuya uniformemente entre las instancias de la aplicación web y móvil, mejorando la disponibilidad y escalabilidad.

- **EC2 (Amazon Elastic Compute Cloud)**:
  - **Descripción**: Servicio de infraestructura como servicio (IaaS) que proporciona capacidad de cómputo escalable en la nube.
  - **Función**: Aloja las instancias de la plataforma core y el sistema complementario que procesan la lógica del negocio y gestionan la base de datos.

- **IAM (Identity and Access Management)**:

- **Descripción**: Servicio de gestión de identidades y accesos en AWS.
- **Función**: Controla y gestiona el acceso a los recursos en AWS mediante la definición de permisos y roles. En este diagrama, se usa para representar un componente de seguridad que gestiona accesos y políticas.

Conexiones y Flujo de Datos

- **User >> Route53 >> CloudFront >> ELB**:
  - **Descripción**: El flujo muestra cómo las solicitudes de los usuarios son gestionadas desde el DNS (Route53), pasando por la CDN (CloudFront), y finalmente balanceadas por ELB hacia las instancias de la aplicación.
- **ELB >> Auth Service**:
  - **Descripción**: ELB dirige las solicitudes a los servicios de autenticación (Auth Service) para verificar la identidad del usuario.
- **Auth Service >> Core System / Supplementary System**:
  - **Descripción**: El servicio de autenticación se comunica con los sistemas core y complementario para procesar las solicitudes del usuario, como consultar movimientos o realizar transferencias.
- **Web App / Mobile App >> Security Service**:
  - **Descripción**: Las aplicaciones web y móvil están conectadas a un servicio de seguridad (IAM como placeholder), que representaría la integración con componentes de seguridad reales como detección de amenazas, inspección de vulnerabilidades, y detección de datos sensibles.

Justificación del Uso de los Componentes

- **Alta Disponibilidad y Escalabilidad**: Usar ELB y CloudFront asegura que el sistema sea capaz de manejar un alto volumen de tráfico y mantener la disponibilidad incluso en caso de fallos en alguna instancia de la aplicación.
- **Seguridad**: El componente genérico de seguridad representa la necesidad de proteger el sistema contra accesos no autorizados y ataques. En una implementación real, esto se traduciría en el uso de servicios específicos para detección de amenazas y gestión de vulnerabilidades.
- **Escalabilidad**: EC2 permite escalar la capacidad de cómputo según las necesidades, asegurando que el sistema pueda manejar un crecimiento en el número de usuarios y transacciones.

Este diagrama proporciona una visión general de cómo los diferentes componentes interactúan en la arquitectura de un sistema de banca por internet, destacando las funciones principales y las conexiones entre los servicios.

Diagrama de Contenedores

El diagrama de contenedores muestra los principales contenedores (aplicaciones, bases de datos, etc.) que componen el sistema y cómo se comunican entre ellos.

Componentes del Diagrama de Contenedores

1. **User (Usuario Final)**
   - **Descripción**: Representa al usuario que interactúa con la aplicación desde un navegador web o una aplicación móvil.
   - **Función**: Inicia solicitudes para acceder a diferentes funciones del sistema, como iniciar sesión, consultar movimientos, y realizar transferencias.
2. **Route53 (DNS y Balanceo de Carga)**
   - **Descripción**: Servicio de DNS y balanceo de carga de AWS.
   - **Función**: Resuelve nombres de dominio y dirige el tráfico a los recursos apropiados. Balancea el tráfico de entrada entre las instancias de la aplicación.
3. **CloudFront (CDN para Entrega de Contenidos)**
   - **Descripción**: Red de entrega de contenidos (CDN) de AWS.
   - **Función**: Distribuye contenido estático (como imágenes, JavaScript y CSS) de manera eficiente a nivel global, reduciendo la latencia y mejorando la velocidad de carga.
4. **ELB (Elastic Load Balancer)**
   - **Descripción**: Servicio de balanceo de carga que distribuye el tráfico entre múltiples instancias de aplicación.
   - **Función**: Asegura que las solicitudes se distribuyan equitativamente entre las instancias para mejorar la disponibilidad y escalabilidad de las aplicaciones web y móviles.
5. **Web App (Aplicación Web SPA)**
   - **Descripción**: Una aplicación web de una sola página (SPA) que proporciona una experiencia de usuario fluida.
   - **Función**: Permite a los usuarios interactuar con el sistema a través de un navegador web. La SPA mejora la velocidad y la interacción del usuario al minimizar las recargas de página.
6. **Mobile App (Aplicación Móvil Multiplataforma)**
   - **Descripción**: Aplicación móvil desarrollada con un framework multiplataforma.
   - **Función**: Proporciona acceso al sistema desde dispositivos móviles, ofreciendo una experiencia de usuario nativa y adaptada a diferentes plataformas.
7. **Auth Service (Servicio de Autenticación)**
   - **Descripción**: Servicio que gestiona la autenticación de usuarios.
   - **Función**: Verifica las credenciales de los usuarios y emite tokens de acceso. Utiliza OAuth 2.0 para garantizar un acceso seguro.
8. **Authz Service (Servicio de Autorización)**
   - **Descripción**: Servicio que gestiona la autorización de acceso a recursos.
   - **Función**: Controla qué recursos y operaciones están disponibles para los usuarios basándose en sus roles y permisos (RBAC).
9. **Transactions Service (Servicio de Transferencias y Pagos)**
   - **Descripción**: Servicio encargado de procesar transferencias y pagos.
   - **Función**: Permite a los usuarios realizar transferencias entre cuentas y procesar pagos. Verifica la disponibilidad de fondos y registra las transacciones.
10. **Movements Service (Servicio de Consulta de Movimientos)**
    - **Descripción**: Servicio que consulta y presenta el historial de movimientos.
    - **Función**: Accede a la plataforma core y al sistema complementario para recuperar el historial de transacciones del usuario y aplicar filtros para obtener información específica.
11. **Notifications DB (Base de Datos de Notificaciones)**
    - **Descripción**: Base de datos que almacena registros de notificaciones enviadas a los usuarios.
    - **Función**: Permite consultar el historial de notificaciones y verificar su estado.

12. **Core DB (Plataforma Core)**
    - **Descripción**: Base de datos que almacena información esencial como clientes, movimientos y productos.
    - **Función**: Proporciona una visión centralizada y detallada de la información financiera y transaccional de los clientes.
13. **Supplementary DB (Sistema Complementario)**
    - **Descripción**: Base de datos que almacena información adicional como direcciones y datos de contacto.
    - **Función**: Completa la información del cliente con datos adicionales necesarios para operaciones detalladas.

Flujo de Datos:

- **User >> Route53 >> CloudFront >> ELB**: El usuario hace solicitudes que son dirigidas por DNS y balanceadas por el ELB.
- **ELB >> Auth Service**: ELB distribuye el tráfico hacia el servicio de autenticación para validar usuarios.
- **Auth Service >> Core DB / Supplementary DB**: El servicio de autenticación consulta las bases de datos para obtener y verificar información.
- **Transactions Service / Movements Service >> Core DB / Supplementary DB**: Los servicios de transacciones y consulta de movimientos acceden a la base de datos core y complementaria para procesar y consultar datos financieros.
- **Transactions Service >> Notifications DB**: El servicio de transacciones almacena registros de notificaciones.

3. Diagrama de Componentes

El diagrama de componentes muestra los componentes dentro de cada contenedor.

Componentes:

1. **User (Usuario Final)**:
    - **Descripción**: Representa al usuario que interactúa con las aplicaciones web o móviles.
    - **Función**: Realiza acciones como iniciar sesión, consultar información y realizar transacciones.
2. **Route53 (DNS y Balanceo de Carga)**:
    - **Descripción**: Amazon Route 53 gestiona el DNS y el balanceo de carga.
    - **Función**: Dirige el tráfico hacia las instancias adecuadas y asegura una distribución equilibrada de solicitudes.
3. **CloudFront (CDN)**:
    - **Descripción**: CDN para la entrega de contenidos de manera eficiente.
    - **Función**: Mejora la velocidad y la experiencia del usuario al entregar contenido desde servidores cercanos.
4. **Web App ELB / Mobile App ELB**:
    - **Descripción**: Elastic Load Balancers específicos para la aplicación web y móvil.
    - **Función**: Distribuyen el tráfico entre las instancias de las aplicaciones web y móviles para asegurar alta disponibilidad y escalabilidad.
5. **Web App / Mobile App**:

    - **Descripción**: Instancias de las aplicaciones web SPA y móvil.
    - **Función**: Proporcionan la interfaz de usuario y la funcionalidad para la interacción con el sistema.
6. **Auth Service (Servicio de Autenticación)**:
    - **Descripción**: Servicio que gestiona la autenticación de usuarios.
    - **Función**: Verifica credenciales y emite tokens de acceso.
7. **Authz Service (Servicio de Autorización)**:
    - **Descripción**: Servicio que gestiona el acceso a recursos basado en roles.
    - **Función**: Controla el acceso a diferentes partes del sistema según los roles de usuario.
8. **Transactions Service (Servicio de Transferencias y Pagos)**:
    - **Descripción**: Servicio que procesa las transferencias y pagos.
    - **Función**: Maneja las transacciones financieras y valida la disponibilidad de fondos.
9. **Movements Service (Servicio de Consulta de Movimientos)**:
    - **Descripción**: Servicio que proporciona información sobre el historial de movimientos.
    - **Función**: Consulta datos de movimientos desde las bases de datos.
10. **Core DB (Plataforma Core)**:
    - **Descripción**: Base de datos que almacena la información central del sistema.
    - **Función**: Almacena datos clave como información de clientes y movimientos financieros.
11. **Supplementary DB (Sistema Complementario)**:
    - **Descripción**: Base de datos adicional que almacena información complementaria.
    - **Función**: Guarda datos adicionales como información de contacto.
12. **Notifications DB (Base de Datos de Notificaciones)**:
    - **Descripción**: Base de datos que gestiona los registros de notificaciones.
    - **Función**: Almacena y consulta el historial de notificaciones enviadas a los usuarios.

Flujo de Datos:

- **User >> Route53 >> CloudFront >> ELB**: Los usuarios acceden a las aplicaciones a través de una CDN y el balanceo de carga asegura que las solicitudes sean distribuidas de manera eficiente.
- **ELB >> Web App / Mobile App**: ELB distribuye el tráfico hacia las instancias de las aplicaciones web y móviles.
- **Web App / Mobile App >> Auth Service**: Las aplicaciones envían solicitudes al servicio de autenticación para verificar usuarios.
- **Auth Service >> Authz Service**: El servicio de autenticación interactúa con el servicio de autorización para controlar el acceso a los recursos.
- **Authz Service >> Transactions Service / Movements Service**: Los servicios de autorización permiten el acceso a los servicios de transacciones y consulta de movimientos.
- **Transactions Service >> Core DB / Supplementary DB / Notifications DB**: El servicio de transacciones y el servicio de consulta de movimientos acceden a las bases de datos para obtener o almacenar información.

Estos diagramas ofrecen una visión detallada de cómo se organizan y conectan los diferentes componentes dentro del sistema de banca por internet, asegurando que cada parte del sistema tenga una función clara y definida.

2. Patrones de Integración y Tecnologías a Utilizar

- **Patrones de Integración**

1. **Microservicios**: Dividir la aplicación en servicios pequeños y especializados que se comunican entre sí mediante API RESTful o mensajes.
2. **Event-Driven Architecture (EDA)**: Utilizar eventos para comunicar cambios y desencadenar acciones en diferentes componentes del sistema bancario. Esto puede incluir el uso de servicios de mensajería como Amazon SQS o Kafka.
3. **API Gateway**: Centralizar el acceso a las API del sistema bancario a través de un único punto de entrada, permitiendo la gestión, seguridad y monitorización centralizadas.
4. **Orquestación de Servicios**: Coordinar la ejecución de múltiples servicios para completar flujos de trabajo complejos dentro del sistema bancario, utilizando herramientas como AWS Step Functions.
5. **Batch Integration**: Intercambiar datos de manera periódica o por lotes entre sistemas bancarios, utilizando técnicas como ETL (Extract, Transform, Load) o procesos de carga por lotes.

Tecnologías Recomendadas

1. **AWS Lambda**: Para la ejecución de funciones sin servidor, que puede integrarse fácilmente con otros servicios de AWS.
2. **Amazon API Gateway**: Para la gestión y exposición segura de API hacia el exterior.
3. **Amazon SQS y SNS**: Para la mensajería y la creación de sistemas basados en eventos.
4. **AWS Step Functions**: Para la orquestación de flujos de trabajo complejos que involucren varios servicios.
5. **AWS EventBridge**: Para la integración basada en eventos entre servicios dentro de la arquitectura de AWS.
6. **AWS AppSync**: Para la creación de APIs GraphQL escalables y seguras, especialmente útiles para aplicaciones móviles y web.
7. **AWS Glue**: Para la preparación y carga de datos a gran escala entre diferentes sistemas bancarios.
8. **Amazon RDS y Amazon DynamoDB**: Para bases de datos relacionales y no relacionales respectivamente, adaptándose a diferentes necesidades de almacenamiento y acceso de datos.
9. **AWS CloudFormation y AWS CDK**: Para la gestión y despliegue automatizado de la infraestructura y recursos en AWS.

3. Requisitos de Seguridad y Cumplimiento Normativo

- **Cumplimiento Normativo**

    - **Regulación Bancaria Local e Internacional:** PCI-DSS, GDPR, PSD2.
    - **Ley Orgánica de Protección de Datos Personales:** Asegurarse de la confidencialidad, integridad y disponibilidad de los datos personales.
- **Seguridad**
    - **Autenticación y Autorización:** OAuth 2.0, OpenID Connect.
    - **Cifrado de Datos:** TLS/SSL para datos en tránsito, AES-256 para datos en reposo.
    - **Monitoreo y Alertas:** Herramientas como Splunk, ELK Stack para SIEM.

Requisitos de Seguridad

1. **Cifrado de Datos**:
    - **En tránsito**: Utilizar HTTPS/TLS para cifrar los datos en tránsito entre los usuarios y los servicios, y entre los servicios internos.
    - **En reposo**: Utilizar AWS KMS para cifrar los datos almacenados en bases de datos, volúmenes de almacenamiento y backups.
2. **Gestión de Identidades y Accesos (IAM)**:
    - Utilizar AWS Cognito para la autenticación y autorización de usuarios.
    - Implementar políticas de IAM estrictas para controlar el acceso a los recursos de AWS.
    - Utilizar roles y políticas basados en el principio de privilegio mínimo.
3. **Monitoreo y Registro**:
    - Utilizar AWS CloudWatch para el monitoreo en tiempo real de los recursos y aplicaciones.
    - Configurar AWS CloudTrail para el registro de auditoría de todas las acciones realizadas en la cuenta de AWS.
    - Implementar AWS Config para asegurar que los recursos cumplan con las configuraciones y políticas definidas.
4. **Seguridad de Red**:
    - Utilizar VPCs para segmentar la red y aislar los diferentes componentes del sistema.
    - Configurar Security Groups y Network ACLs para controlar el tráfico entrante y saliente.
    - Utilizar AWS Shield y AWS WAF para protegerse contra ataques DDoS y amenazas web.

Cumplimiento Normativo

1. **Ley Orgánica de Protección de Datos Personales (LOPDP)**:
    - **Consentimiento**: Implementar mecanismos para obtener y gestionar el consentimiento explícito de los usuarios para el tratamiento de sus datos personales.
    - **Derecho de Acceso y Rectificación**: Proveer a los usuarios la capacidad de acceder y corregir sus datos personales.
    - **Portabilidad de Datos**: Implementar funcionalidades que permitan a los usuarios exportar sus datos personales en un formato estructurado, comúnmente utilizado y legible por máquina.
    - **Retención de Datos**: Definir y aplicar políticas de retención y eliminación de datos personales, asegurando que los datos no se conserven más tiempo del necesario.

2. **Cumplimiento de Normativas Financieras**:
    - **PCI DSS**: Si se procesan pagos con tarjetas de crédito, cumplir con los requisitos del estándar de seguridad de datos de la industria de tarjetas de pago (PCI DSS).
    - **SOX**: Cumplir con la Ley Sarbanes-Oxley, implementando controles internos sólidos para garantizar la precisión y fiabilidad de los informes financieros.

4. Estrategia para Alta Disponibilidad y Recuperación ante Desastres

- **Alta Disponibilidad (HA)**
    - **Clústeres Multi-región:** Implementar servicios en múltiples zonas de disponibilidad.
    - **Balanceo de Carga:** Utilizar balanceadores de carga para distribuir el tráfico. Ejemplo: AWS ELB.
- **Recuperación ante Desastres (DR)**
    - **Backups Regulares:** Copias de seguridad frecuentes y automáticas.
    - **Sitios de Recuperación:** Tener un sitio secundario de recuperación en otra región geográfica.

Explicación de Componentes Adicionales:

1. **Multi-AZ (Alta Disponibilidad):** Las bases de datos tradicionales y digitales del core bancario están configuradas para utilizar Multi-AZ en RDS para garantizar la alta disponibilidad y tolerancia a fallos.
2. **Auto-scaling (Alta Disponibilidad):** Los servicios de web banking, mobile banking, payment services, risk management y fraud prevention utilizan ECS con auto-escalado para garantizar que la capacidad se ajuste a la demanda.
3. **Load Balancer (Alta Disponibilidad):** Un balanceador de carga (ELB) distribuye el tráfico entrante entre las instancias de EC2, garantizando una distribución uniforme y reduciendo el riesgo de sobrecarga.
4. **Monitoring & Logging (Seguridad y Monitoreo):** Cloudwatch para el monitoreo continuo del sistema, Cloudtrail para el registro de auditoría y AWS Config para la supervisión de configuraciones y cumplimiento normativo.
5. **Data Encryption (Seguridad):** KMS se utiliza para cifrar los datos en reposo y en tránsito, garantizando la protección de la información sensible.
6. **Message Broker (Tolerancia a Fallos):** SQS se utiliza como un intermediario de mensajes para asegurar la comunicación entre los sistemas, manejando fallos y garantizando la entrega.

Consideraciones adicionales para Implementación:

1. **Alta Disponibilidad y Tolerancia a Fallos:**
    - Implementa Multi-AZ para todas las bases de datos críticas.
    - Configura auto-escalado para los servicios críticos.
    - Utiliza balanceadores de carga para distribuir el tráfico y garantizar la alta disponibilidad.
2. **Seguridad:**
    - Implementa cifrado de datos con KMS.
    - Utiliza AWS Cognito para la gestión de identidades y accesos.
    - Configura monitoreo y registros de auditoría con Cloudwatch y Cloudtrail.
3. **Monitoreo:**
    - Configura AWS Config para la supervisión de configuraciones y cumplimiento normativo.
    - Implementa dashboards y alarmas en Cloudwatch para el monitoreo en tiempo real.

Estos ajustes garantizan que el sistema no solo sea altamente disponible y tolerante a fallos, sino también seguro y monitorizado adecuadamente.

6. Gestión de Identidad y Acceso

- **Sistema de Gestión de Identidades (IAM):** Implementar un IAM robusto que soporte MFA, SSO.
- **Roles y Permisos Granulares:** Definir roles y permisos detallados para limitar el acceso basado en el principio de menor privilegio.

7. Estrategia de API Internas y Externas

- **APIs Internas:** Utilizar patrones de diseño como API Gateway y Service Mesh para gestionar la comunicación interna.
- **APIs Externas:** Implementar estándares como REST y GraphQL, con controles de seguridad como rate limiting y JWT.

8. Modelo de Gobierno de APIs y Microservicios

- **Catálogo de APIs:** Documentar y catalogar todas las APIs disponibles.
- **Versionamiento de APIs:** Implementar un sistema de versionamiento robusto.
- **Monitoreo y Logging:** Herramientas para monitoreo y logging de APIs y microservicios. Ejemplo: Prometheus, Grafana.

9. Plan de Migración Gradual

- **Migración Faseada:** Dividir la migración en fases manejables, comenzando por servicios menos críticos.
- **Pruebas y Validación:** Realizar pruebas exhaustivas en cada fase antes de la migración completa.

- **Backups y Rollbacks:** Tener siempre un plan de reversión en caso de fallos durante la migración.

Justificación de la Propuesta

- **Mejoras en la flexibilidad y escalabilidad:** La adopción de microservicios y APIs permite una evolución tecnológica más ágil y escalable.
- **Cumplimiento Normativo y Seguridad:** La arquitectura propuesta asegura el cumplimiento con regulaciones y estándares de seguridad.
- **Alta Disponibilidad y Recuperación ante Desastres:** La implementación de estrategias HA y DR garantiza la continuidad del negocio.
- **Experiencia del Usuario:** La modernización mejora significativamente la experiencia del usuario final con servicios más rápidos y fiables.

Alta Disponibilidad (HA)

1. **Implementación de Multi-AZ (Availability Zones)**:
    - **Bases de Datos**: Configurar las bases de datos en modo Multi-AZ, lo que permite replicar los datos en diferentes zonas de disponibilidad. En caso de fallo en una zona, la base de datos puede conmutar por error a la réplica en otra zona.
    - **Servidores de Aplicaciones**: Desplegar los servidores de aplicaciones (web, móvil, pagos, APIs, etc.) en múltiples zonas de disponibilidad para asegurar que una falla en una zona no afecte la disponibilidad general del servicio.
2. **Load Balancing (Balanceo de Carga)**:
    - Utilizar ELB (Elastic Load Balancer) para distribuir el tráfico entre múltiples instancias de servidores de aplicaciones en diferentes zonas de disponibilidad, garantizando la disponibilidad y optimizando el rendimiento.
3. **Auto-scaling (Autoescalado)**:
    - Configurar grupos de autoescalado para las instancias de servidores de aplicaciones, lo que permite aumentar o disminuir automáticamente el número de instancias en función de la demanda, asegurando así la disponibilidad y eficiencia de los recursos.
4. **Redundancia en la Infraestructura de Red**:
    - Implementar redundancia en las conexiones de red y en los componentes críticos de la infraestructura para evitar puntos únicos de fallo.

Recuperación Ante Desastres (DR)

1. **Backups y Recuperación de Datos**:
    - Configurar backups automáticos y regulares de todas las bases de datos y volúmenes de almacenamiento utilizando servicios como AWS Backup o snapshots de RDS.

    - Probar regularmente los procedimientos de restauración de backups para asegurar que los datos pueden ser recuperados de manera efectiva en caso de desastre.
2. **Planes de Recuperación Ante Desastres**:
    - Desarrollar y documentar un plan de recuperación ante desastres que incluya procedimientos detallados para restaurar sistemas y datos en caso de fallo catastrófico.
    - Realizar pruebas regulares del plan de recuperación para identificar y corregir posibles fallos o debilidades.
3. **Replicación entre Regiones**:
    - Configurar la replicación de datos críticos entre diferentes regiones geográficas para asegurar que, en caso de un desastre que afecte a una región completa, los datos y servicios puedan ser restaurados en otra región.
4. **Configuración de Sitio de Recuperación**:
    - Establecer un sitio de recuperación (DR site) en una región diferente, configurado para asumir rápidamente las operaciones en caso de que el sitio principal falle. Este sitio puede ser configurado en modo activo-activo o activo-pasivo, dependiendo de los requisitos específicos de recuperación.

Beneficios:

La incorporación de componentes de gestión de riesgos y fraude en la arquitectura proporciona los siguientes beneficios:

- **Mayor seguridad:** Protege el sistema contra amenazas maliciosas, vulnerabilidades y fugas de datos.
- **Cumplimiento normativo:** Ayuda a cumplir con las regulaciones de seguridad de datos y protección financiera.
- **Confianza del cliente:** Aumenta la confianza de los usuarios en la seguridad y protección de sus datos.

Conclusiones:

Este diagrama ilustra una arquitectura de banca por internet robusta y segura que incorpora medidas para la gestión de riesgos y la prevención del fraude. Al implementar estos componentes, las instituciones financieras pueden proteger sus sistemas, cumplir con las regulaciones y mejorar la confianza de sus clientes.

Se adjuntan los diagramas de los diseños de arquitectura.