BANCO PICHINCHA

Reto

Arquitectura de Soluciones.

Descripción

Usted ha sido contratado por una entidad llamada BP como arquitecto de soluciones para diseñar un sistema de banca por internet, en este sistema los usuarios podrán acceder al histórico de sus movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias.
Toda la información referente al cliente se tomará de 2 sistemas, una plataforma Core que contiene información básica de cliente, productos, cuentas, movimientos, y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle este sistema core está conectado a una base de datos principal.

Debido a que la norma exige que los usuarios sean notificados sobre los movimientos realizados, el sistema utilizará sistemas externos o propios de envío de notificaciones, mínimo 2.

Este sistema contará con 2 aplicaciones en el Front, una SPA y una Aplicación móvil desarrollada en un Framework multiplataforma. (Mencione 2 opciones y justifique el porqué de su elección)

Ambas aplicaciones autenticarán a los usuarios mediante un servicio que usa el estándar Oauth2.0, para el cual no requiere implementar toda la lógica, ya que la compañía cuenta con un producto que puede ser configurado para este fin; sin embargo, debe dar recomendaciones sobre cuál es el mejor flujo de autenticación que se debería usar según el estándar.

Tenga en cuenta que el sistema de Onboarding para nuevos clientes en la aplicación móvil usa reconocimiento facial, por tanto, su arquitectura deberá considerarlo como parte del flujo de autorización y autenticación, a partir del Onboarding el nuevo usuario podrá ingresar al sistema mediante usuario y clave, huella o algún otro método especifique alguno de los anteriores dentro de su arquitectura, también puede recomendar herramientas de industria que realicen estas tareas y robustezca su aplicación.

El sistema utiliza usa base de datos de auditoría que registra todas las acciones del cliente y cuenta con un mecanismo de persistencia de información para clientes frecuentes, para este caso proponga una alternativa basada en patrones de diseño que relacione los componentes que deberían interactuar para conseguir el objetivo.

Para obtener los datos del cliente el sistema pasa por una capa de integración compuesta por un api Gateway y consume los servicios necesarios de acuerdo con el tipo de transacción,

inicialmente usted cuenta con 3 servicios principales, consulta de datos básicos, consulta de movimientos y transferencias, que realiza llamados a servicios externos dependiendo del tipo, si considera que debería agregar más servicios para mejorar la repuesta de información a sus clientes, es libre de hacerlo.

Consideraciones.

A) Para este reto, mencione aquellos elementos normativos que podrían ser importantes a la hora de crear aplicaciones para entidades financieras, Ejemplo ley de datos personales, seguridad etc.

b) Garantice en su arquitectura, alta disponibilidad (**HA**), tolerancia a fallos, recuperación ante desastres (**DR**), Seguridad y Monitoreo, Excelencia operativa y auto-healing.

c) Sí lo considera necesario, su arquitectura puede contener elementos de infraestructura en nube como Azure o AWS, garantice baja latencia en sus servicios, cuenta con presupuesto para esto.

d) En lo posible plantee una arquitectura desacoplada con elementos y reusables y cohesionados para otros componentes que puedan adicionarse en el futuro.

El modelo debe ser desarrollado bajo **c4**(Modelo de Contexto, Modelo de aplicación o contenedor y Componentes), describa hasta el modelo de componentes, la infraestructura la puede modelar como usted lo considere usando la herramienta de su preferencia.

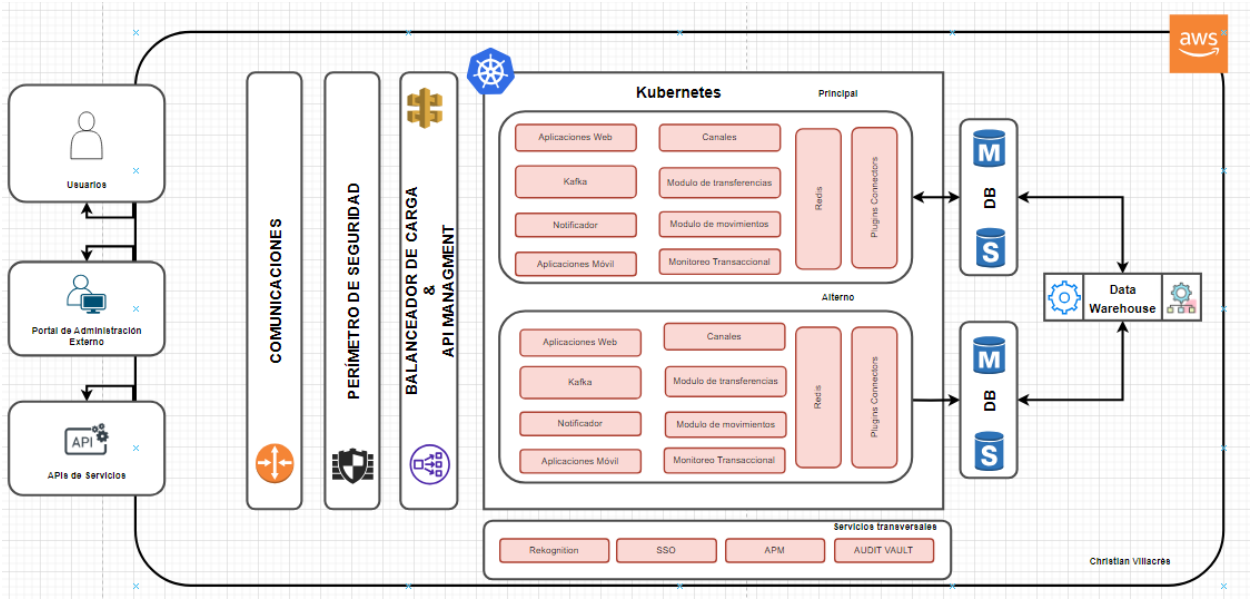
¡Éxitos!

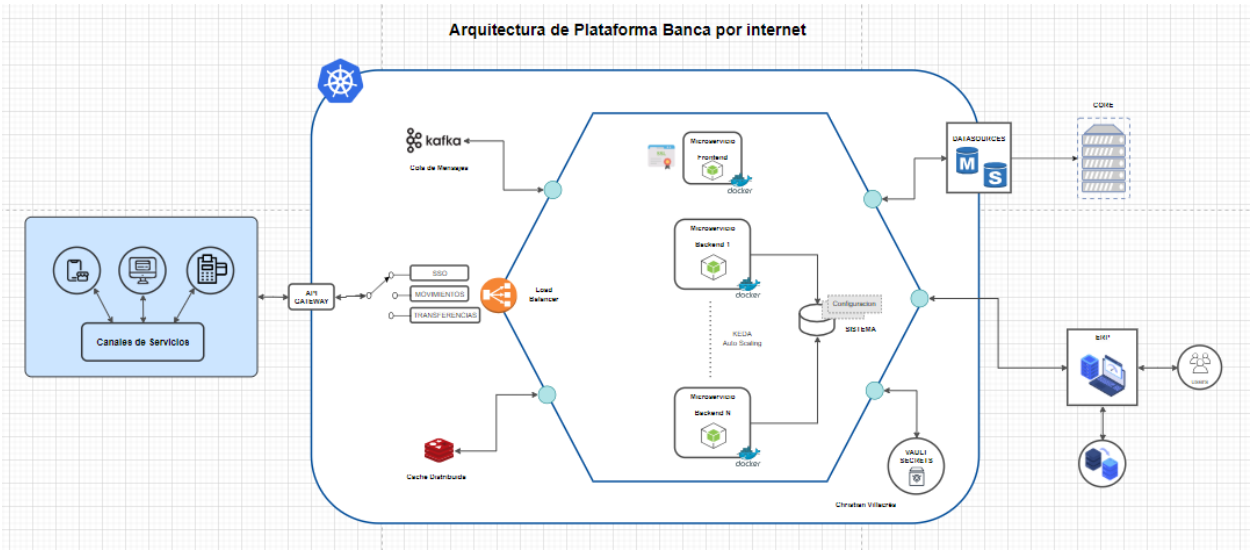
Solución:

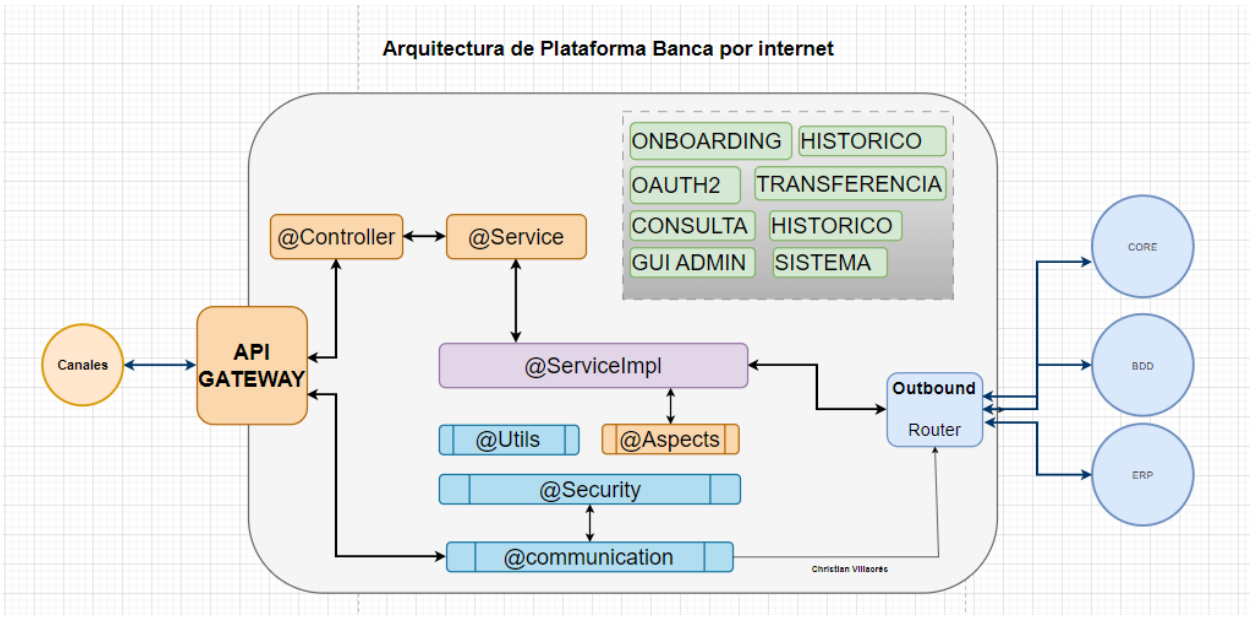
Arquitectura basada en microservicios y contenedores
Elementos primarios:
2 sistemas
- 1 core
- 1 erp
- BDD

1 notificador - 1 componente de colas
1 microservicio UI (angular o react) porque son arquitecturas SPA (single page application) y poseen un eficiente renderizado y extensibilidad.
Flutter o Ionic (compatibilidad IOS o ANDROID) por tiempos, costos, compatibilidad multiplataforma y mantenimiento.
Perímetro de seguridad (DMZ, IPS, WAF, FIREWALLS, PROXYS).
API MANAGER (mTls, certificados, control de cabeceras, firmas de mensajes, federación de usuarios)
1 componente con Single SingOn utilizar servicios de reconocimiento de cloud como cognito y Rekognition para validación de sistemas biométricos.
Posterior al API MANAGER debe existir un balanceador (ejemplos: F5, NGINX, APACHE).
Para tolerancia a fallos la solución debe estar en Kubernetes u Openshift que permiten realizar escalabilidad de manera automática y controles de resiliencia.
Para recuperación de desastres (DR) de acuerdo con los valores de RPO y RTO se recomienda que la solución sea activa/activa en dos sitios un principal y un alterno.
Debe existir un APM como Dynatrace, Grafana o DataDog,
Las aplicaciones en la nube deben ser multiregión y dependiendo de la necesidad de la solución y las políticas de la organización, se puede realizar un esquema Hibrido, haciendo una nube hibrida privada, en la nube los servicios de APIGATEWAY, APIMANAGER, Servicios de seguridad, Sandboxs y en OnPremise, Bases de datos, ERP etc. Y entre ambas un proceso de sincronización.
Patron de diseño saga o CQRS.

Diagrama propuesto:






Arquitectura de Plataforma Banca por internet
ONBOARDING
HISTORICO
OAUTH2
TRANSFERENCIA
CONSULTA
HISTORICO
GUI ADMIN
SISTEMA
@Controller
@Service
Canales
API GATEWAY
@ServiceImpl
@Utils
@Aspects
@Security
@communication
Outbound
Router
CORE
BDD
ERP
Christian Villacrés