Reto Técnico

Principales problemas y desafíos en Reto Técnico que he agrupado en 4 grupos.

- Nomativas
- Seguridad
- Diseño Banca Virtual
- Excelencia operativa y auto-healing.
- Integración

Para los diagramas 4C se dividen en 3 enfoques que son.

1. Diagrama Contexto, énfasis en actores, canales y servicios Bancarios.
2. Diagrama Contenedores, énfasis en los productos y servicios bancarios.
3. Diagrama Componentes, énfasis en las justificaciones técnicas.

Normativas

- Se debe tomar en cuenta las normativas del sector financiero local e internacional.

Normativas Nacionales

- LOPDP (Ley Orgánica de Protección de Datos Personales)
- SB-21-2126 Riesgo Operativo
- SB-2016-940 Sobre las Transacciones Financieras y Seguridad de la Información
- ADM-2014-12708 Sobre la Infraestructura tecnológica en Entidades Financieras.

Normativas y Estándares Internacionales

- ISO 2700 Estándar de Seguridad de Sistemas Informáticos.
- PCI DSS Estándar de Seguridad de Datos para la Industria de Tarjetas de Pago
- GDPR (General Data Protection Regulation)
- PSD2 (Payment Services Directive2) Normativa para proteger al consumidor de transacciones no autorizadas.
- TPP (Standard Third-Party Provider)
- XS2A (Standard Access to Accounts)
- Modelo COSO Control para auditoria financiera.

Seguridad

- Sistema Onboarding para App Mobile
  - Reconocimiento facial
- Autenticación y Autorización para Banca Virtual
  - Huella y clave
  - Oauth2
- Base de Datos de Auditoría para las acciones del cliente.
- Las transacciones deben asegurar los pilares de la información Confidencialidad, Integridad, disponibilidad y no repudio.

Soluciones Técnicas Propuestas

Onboarding con Reconocimiento Facial

- Id-Check de RecFace servicio biométrico permite verificar la identidad de una persona mediante fotografía y se puede integrar en aplicaciones móviles, es usada entidades de crédito.
- Proporciona una API para la integración con sistemas de terceros o la incorporación de datos biométricos en App Mobile.

Autenticación y Autorización

- SCA (Strong Customer Authentication) requisito del PSD2 que utiliza MFA (Multifactor Authentication)
- Para Oauth2 se puede usar Okta o Auth0 que ofrecen SDK y APIs para integrar a la Banca Web o a Banca Mobile.
- Los usuarios se deben administrar por medio de un LDAP de clientes, independiente del LDAP de usuarios.
- Para el reconocimiento de huella se puede usar la librería react-native-biometrics de Native React.

Base de Datos de Audiroría

- Se puede usar un Modelo Event Sourcing donde cada cambio en el estado del sistema se almacena en registro de eventos en una BDD NoSql ACID.
- Esta base puede ser consumida por un Sistemas de Monitorie Transaccional para detectar actividades sospechosas o patrones de fraude, que están conectados a un sistema de notificación al cliente.

Sistema de Notificación

- Se puede usar un sistema multicanal (WhatsApp, SMS, eMail, mensajes de voz) SaaS ej. Twilio, conectado al TSM, también al flujo transaccional mediante llamadas asíncronas.
- Bajo nuestro control se puede usar el servicio de notificaciones de nuestra App Mobile sin que la transacción se haya ejecutado en la misma canal.
- Esto se puede complementar con transacciones con retardos en el asiento transaccional y su posterior mensaje de efectivicación.

Diseño Banca Virtual

- El sistema debe contar con 2 Aplicaciones Front:
  - App Mobile (Banca Móvil)
    - Consulta movimientos
    - Onboarding
    - Notificaciones
  - App Web (Banca Web)
    - Consulta de Movimientos.
- Modelo de persistencia para clientes frecuentes.

Soluciones Técnicas Propuestas

- Se recomienda un software multiplataforma como puede ser Native React y Xamarine Existen nuevas opciones con Fluttel y Kotlin.
- La interface grafica debe estar diseñadas con UI/UX para mejor experiencia del usuario.
- Debe estar diseñada con una interface responsiva ej. Bootstrap para que sea independiente del tamaño del marco de la ventana y de su resolución
- Los consumos de servicios deben ser a través llamadas a un Api Gatway que se encargue de la seguridad de las transacciones.
- El Ali Gayway expone a microservicios que a sus vez puede interactuar con otros microservicios o App Legadas o Core Banking a través el ESB.
- Persitencia para los Clientes frecuentes se puede usar una BDD In Memory como Redis

Excelencia operativa y auto-healing.

- Los sistemas transaccionales bancarios deben estar disponibles 24/7.

Soluciones Técnicas Propuestas

- Para Topología On Premise se debe tener un Data Center Alterno con más de 40Km de distancia, en regiones diferentes con fibra oscura de baja latencia para replicas.
- Para aplicaciones monolíticas se usa varios mecanismos, servidores con replicas en un modelo Activo/Pasivo, con Ips flotantes en caso de falla.
- Para Base de Datos se puede usar un cluster de base de datos con replicas, procesos de backup y restore.
- Para Aplicaciones Legadas se puede usar Clustering y Balanceadores de Carga en caso de aplicaciones stateless y NBL o Cahe distribuida para Stateful.
- Para microservicios gestionados se puede usar Auto Scaling.
- En el caso de servicios en la nube se puede tener replicas en regiones diferentes y un DNS distribuido.
- El negocio debe tener un Stand in para transacciones con márgenes personalizados por cluster de clientes y canales.

Integración

- Integración sobre varios tipos de plataformas, sistemas, tecnologías y arquitecturas.
  - Consulta cliente:
    - plataforma Core (Core Bancario Monolítica).
    - otros servicios de clientes (CRM SaaS).
  - Consulta movimientos:
    - históricos movimientos propios (afectando las cuentas del Core Bancario).
    - Movimientos interbancarios (a través Redes Bancarias).
  - Api Gatewate para exponer servicios a (Banca Virtual)

Soluciones Técnicas Propuestas

Datos Cliente

- Datos Core Banking normalmente está desarrollado usando arquitectura monolíticas donde se usa interfaces protocolos conectados como son: CICS, RPC, Corba, Socket TCP, Rmi, Com, etc, para lo cual se usa ESB que tiene la capacidad de traducir mediante conectores estos protocolos desconectados para exponer API Rest y SOAP que pueden ser consumidos por Legacies, Microservicios o directamente por un Api Gateway.
  Por buenas prácticas arquitectónicas no es recomendable el consumo directo de la BBD
  No recommendable: Api Gateway →Microservice→ BDD Core Banking
  Aunque si se podría hacer en el caso de una BDD espejo de lectura.
  Recommendable 1: Api Gateway →Microservice→ Replica BDD Core Banking
  Para la consulta de clientes podría usar una integración
  Recommendable 2: Api Gateway →ESB→ Modulo Clientes Core Banking
- Para los datos extras de cliente se usa una plataforma CRM que normalmente exponen conectores API, convirtiendo una integración sencilla:
  Api Gateway → API CRM
  La recomendación más acertada seria que tanto para consultas cortas como para consultas detalladas se use solamente el CRM.

Transacciones Red Bancaria

El banco BP expone y consume servicios a las Redes Bancarias las mismas que técnicamente puede ser de la siguiente forma:

- Para exponer servicios con Franquicias se unas una App Gatewate para recibir transacciones TCP Iso8583 ej. MPGS y VPP o BanRed. Las mismas que exponen APIs, para transferencias Swift se unas archivos de mensajes en repositorios seguros que luego entran a una cámara de compensación por procesos batch. Para lo que son Wallets o Fintch se expone por medio de un Api Gateway.
- Para invocar o consumir servicios se puede usar App Proxys o usar un ESB para protocolos que no sean del tipo API Rest.
- Todas las transacciones quedan en registros de logs seguros (revisados en el grupo anterior)

Consulta de movimientos

- Todas las transacciones financieras locales o externas quedan registradas en el Core Banking y son expuestas a la Banca Virtual a través del Api Gateway de la siguiente manera.
  Gateway →ESB→ Modulo Cuentas Core Banking

1. Diagrama de Contexto

En el diagrama de Contexto divido por capas los sistemas, donde pongo énfasis en la parte normativa de actores, canales y servicios Bancarios.

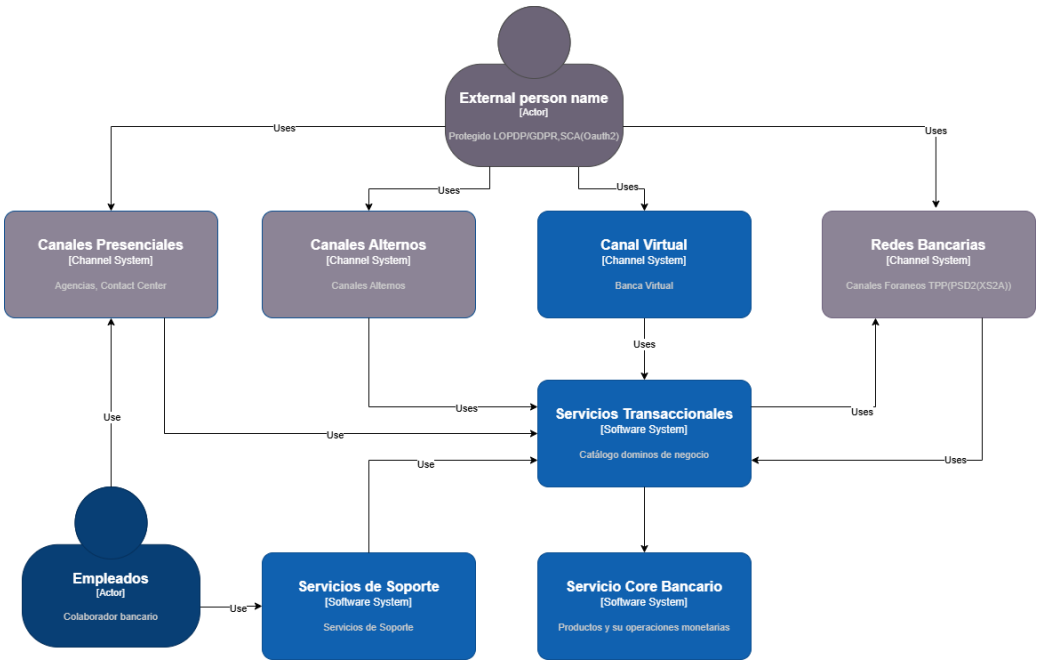

Diagrama de Contexto

1.1. Actores

1.1.1. Cliente

Quién es dueño de los productos bancarios y realiza las operaciones bancarias usando los servicios bancarios.

1.2. Canales Bancarios

1.2.1. Canal Virtual

Banca Mobile y Web (Banca Internet).

1.2.2. Canales Alternos

Canales electrónicos: Atm propios, PoS Propios, Ivr, Voicebot, Chatbot, Kioskos, etc.

1.2.3. Redes Bancarias

Redes de servicios bancarios asociados como son: Franquicias de tarjetas de crédito, Remesadoras, Transferencias Swift, Sistema Interbancario del BCE, B2B (Businesses to Businesses) con distintas plataformas de cobro o servicios de cobro a instituciones, Fintech como son billeteras móviles y pagos en línea.

1.2.4. Canales Presenciales

Agencias y Contact Center donde el cliente tiene una interrelación personalizada.
(Este canal no se lo tratará ya que no se requiere para el ejercicio)

1.3. Servicios Bancarios

1.3.1. Servicios Transaccionales

Son las distintas interfaces que exponen servicios bancarios que se pueden agrupar en dominios de negocio.

1.3.2. Servicios de Soporte

Son las plataformas de soporte a los productos y servicios financieros.

1.3.3. Servicio Core Banking

Es el sistema central de los productos bancarios, y todas las operaciones transaccionales sobre los mismos.

2. Diagrama de Contenedores

En el diagrama de Contenedores pongo énfasis en los productos y servicios bancarios.

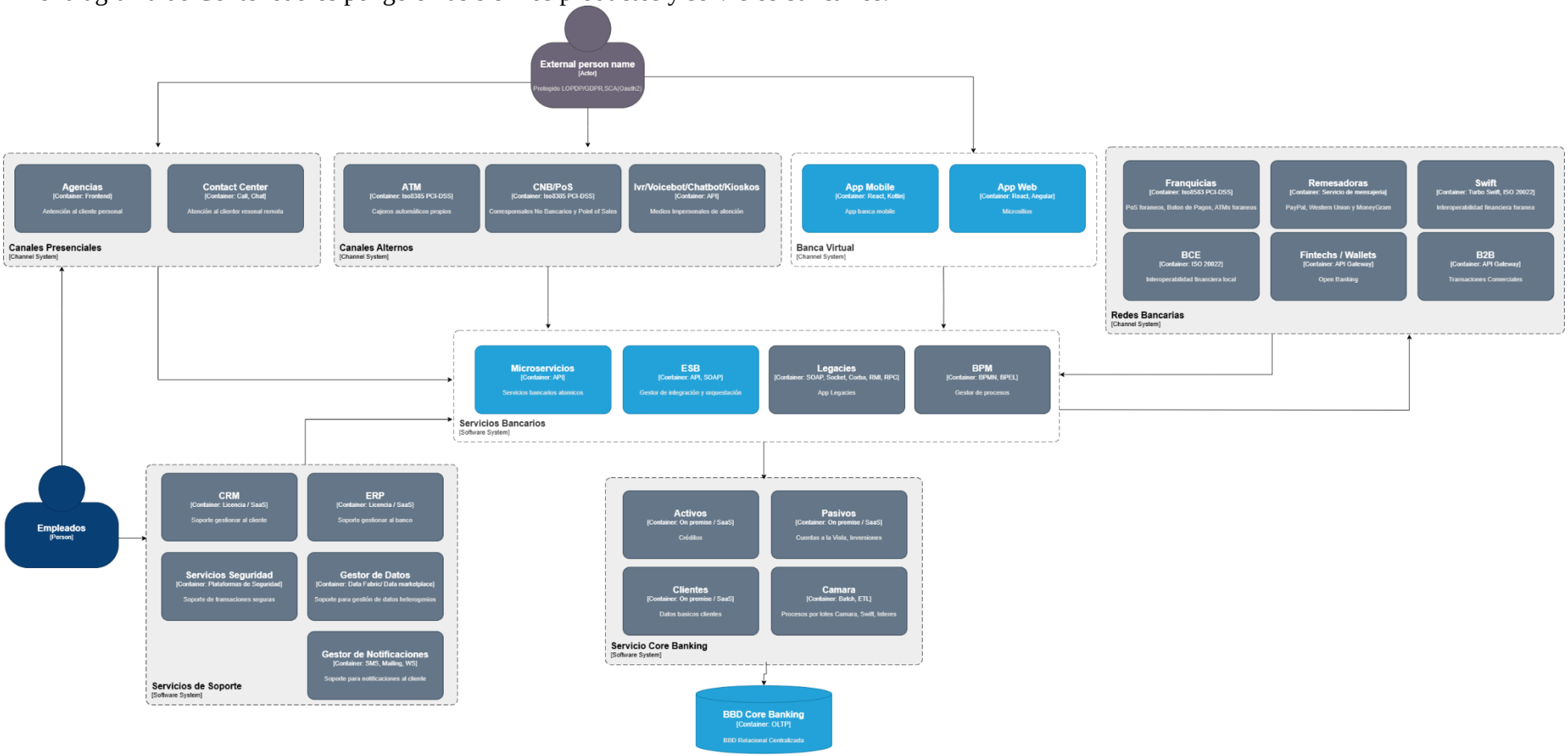

Diagrama de Contenedores

1.1. Banca Virtual

1.1.1. Banca Mobile

Se tiene los servicios de transferencias,

Dominios

Onboarding, gestión de cuentas, préstamos y créditos, inversiones, transferencias, pagos, consulta de movimientos,

1.1.2. Banca Web (Internet)

Se tiene los servicios de transferencias,

Dominios

Gestión de cuentas, préstamos y créditos, inversiones, transferencias, pagos, consulta de movimientos.

1.2. Servicios Bancarios

Capa intermedia de Integración entre servicios transaccionales, servicios de soporte y redes externas.

1.2.1. Microservicios

Se tiene los servicios atómicos, que cumplen puntualmente un propósito con el mínimo de acoplamiento y alta cohesión con toros servicios en una arquitectura de llamadas por eventos, cada microservicio garantiza las propiedades ACID de su propia data.
Generalmente se desarrollan microservicios cuando se crea un nuevo servicio bancario.

1.2.2. Enterprise Service Bus

Es una capa de integración donde se orquesta diferentes tipos de servicios (BPEL) con distintos tipos de protocolos, CISC, RPC, COM, RMI, CORBA, SOAP, API.
Una de sus principales características es que maneja a través de MQ, colas de mensajes para transacciones asíncronas.

1.2.3. App Legacies

Hay muchas aplicaciones, y plataformas que conviven con servicios modernos, que no se han podido migrar por su confiabilidad y/o su complejidad. Pero que todavía son usadas y necesitan integrarse con otros legacies o micros servicios modernos.

1.2.4. BPM

El Gestión de Procesos de Negocio crea flujos BPMN donde se automatiza ciertos procesos de negocio. El BPM se lo puede considerar una aplicación legacy ya que RPA, ML, IA y plataformas Low Code, están remplazando.

1.3. Servicios de Soporte

Capa intermedia de servicios no transaccionales que pan soporte a las transacciones o a servicios no transaccionales.

1.3.1. Servicios de Seguridad

Posiblemente el servicio de no transaccional más importante, que permite garantizar la confiabilidad, integridad y disponibilidad de la información. Donde existen servicios como el de autenticación, validación biométrica, control, monitoreo de aplicaciones, etc

1.3.2. CRM

El gestión de relaciones con clientes es una pieza clave para el negocio, para campañas de productos y servicios bancarios donde de clusteriza a los clientes para brindar un mejor servicio de calidad.

1.3.3. Gestor de Notificaciones

La gestión de notificación es muy importante para que entre otras cosas se informe a tiempo sobre una suplantación de identidad o una transacción fraudulenta.

3. Diagrama de Componentes

En el diagrama de Componentes donde pongo énfasis en las justificaciones técnicas.

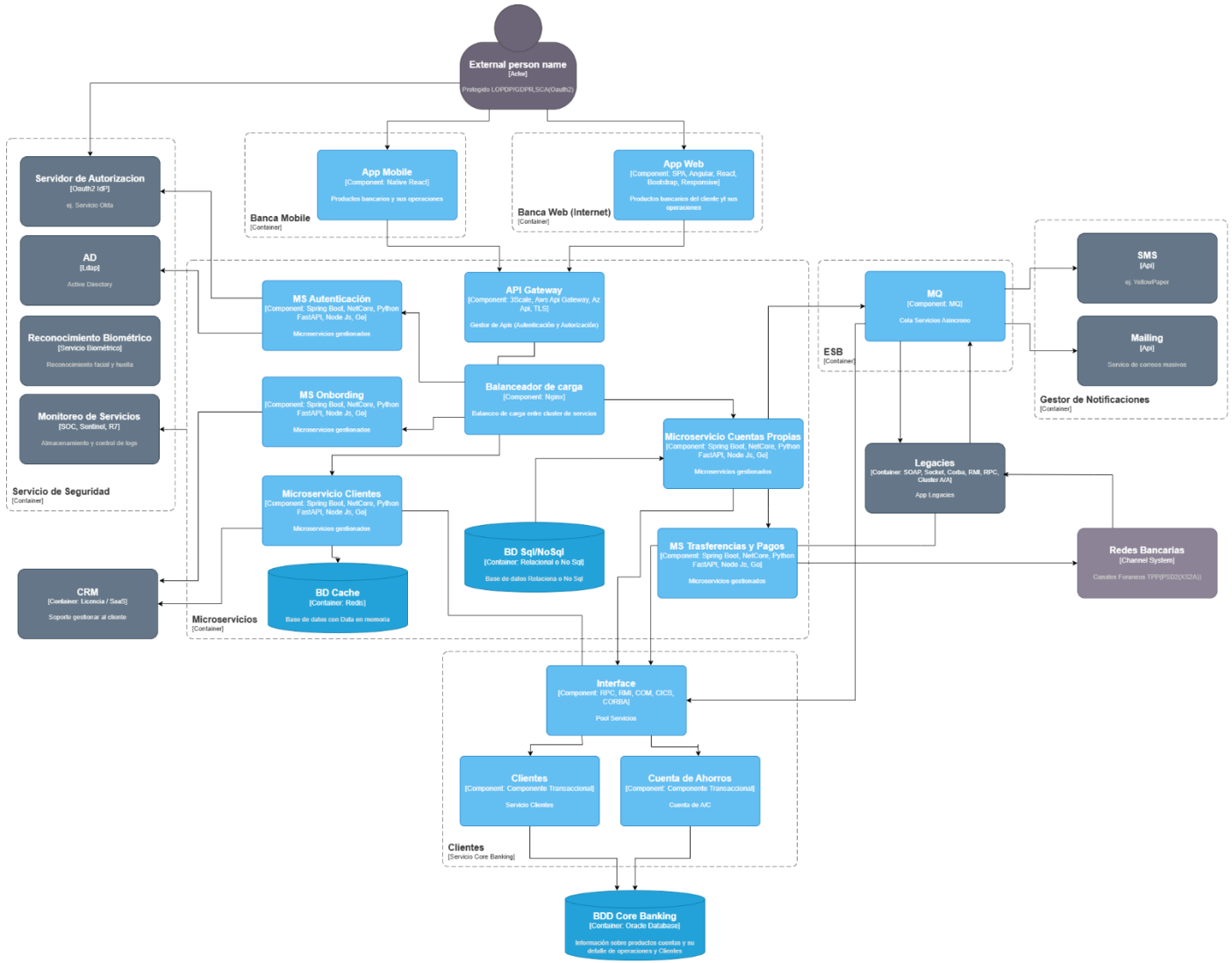

Diagrama de Componentes

1.1. Servicios SaaS

Se recomienda para servicios que no son parte del giro de negocio o donde la información no es confidencial usar servicios SaaS, en el ejercicio planteado se puede contratar servicios de validación biométrico, de notificación masiva de correos y mensajes de texto.

1.2. Tecnologías Front-end

1.2.1. App Mobile

La aplicación mobile, recomiendo Native React ya que es independiente de la plataforma sirve tanto para Android, iOS, Windows, macOS, Linux, para cualquier navegador Web.
La alta disponibilidad se la puede manejar mediante un Api Gateway distribuido en 2 regiones diferentes. También existen mecanismos de negocio como el Stand-In para manejo de cupos, para usuarios que necesitan hacer transacciones fuera de línea.

1.1.1. App Web (Internet)

Se usaría tecnología SPA, responsive web design, con una arquitectura micrositio se puede usar varios tiposde librerías para sacar el mejor provecho de cada una: React, Angular, Vue,js, Django.
La alta disponibilidad se la maneja con sitios en al menos 2 regiones, con micrositios que manejen servicios elásticos.

1.3. Tecnologías Middleware

1.3.1. Micro servicios

Los micro servicios deben exponer API Rest, se puede construir con frameworks livianos como son Spring Boot, Micronaut con Kotlin, Go, en lo particular me gusta Fast API con Python.
El acceso a datos se puede usar una arquitectura CQRS (Command Query Responsibility Segregation) para mejorar los tiempos
Crear una arquitectura orientada a evento, para mantener desacoplada la interacciones.
Usar mecanismos asíncronos para llamadas a servicios de baja latencia en el tiempo de respuesta de la transacción financiera.
La alta disponibilidad se puede manejar con balanceadores tipo Nginx y microservicios gestionados con crecimiento elástico automático.

1.4. Back-End

1.4.1. Base de Datos Relacionales

Base de datos ligeras como PostgrSql, MySql, para el core bancario puede ser Oracle o SqlServer con arquitectura Mirror para la alta disponibilidad,

1.4.2. Base de Datos in Memory

Redis es la más popular, para datos frecuentes de Clientes.

1.1.1. Base de Datos NoSql

Base de Datos Mongo DB

1.1.2. ETLs

Manejo de datos con Az Data Factory y AWS Data Pipe Line

1.1.3. Gestion de Datos

Exposición de datos Snowflake, Data Fabric

1.1.4. Gestion de Datos de Auditoría

Sentinel y Data Monitor de Azure es la opción que más conozco.

1.1.5. Core Banking

La alta disponibilidad para un Core Banking on Premise monolitico se puede manejar con un esquema Activo/Pasivo.