Para abordar el escenario planteado, se realizan diagramas C4 con la solución propuesta:

URL del repositorio:

https://github.com/diegobsc23/testarqint/

Diagrama de Contexto:

En este diagrama se muestra como el Cliente utiliza el Sistema Bancario el cual a su vez se integra con sistemas externos como Entidades Regulatorias y Servicios/APIs de Terceros para el pago de servicios.

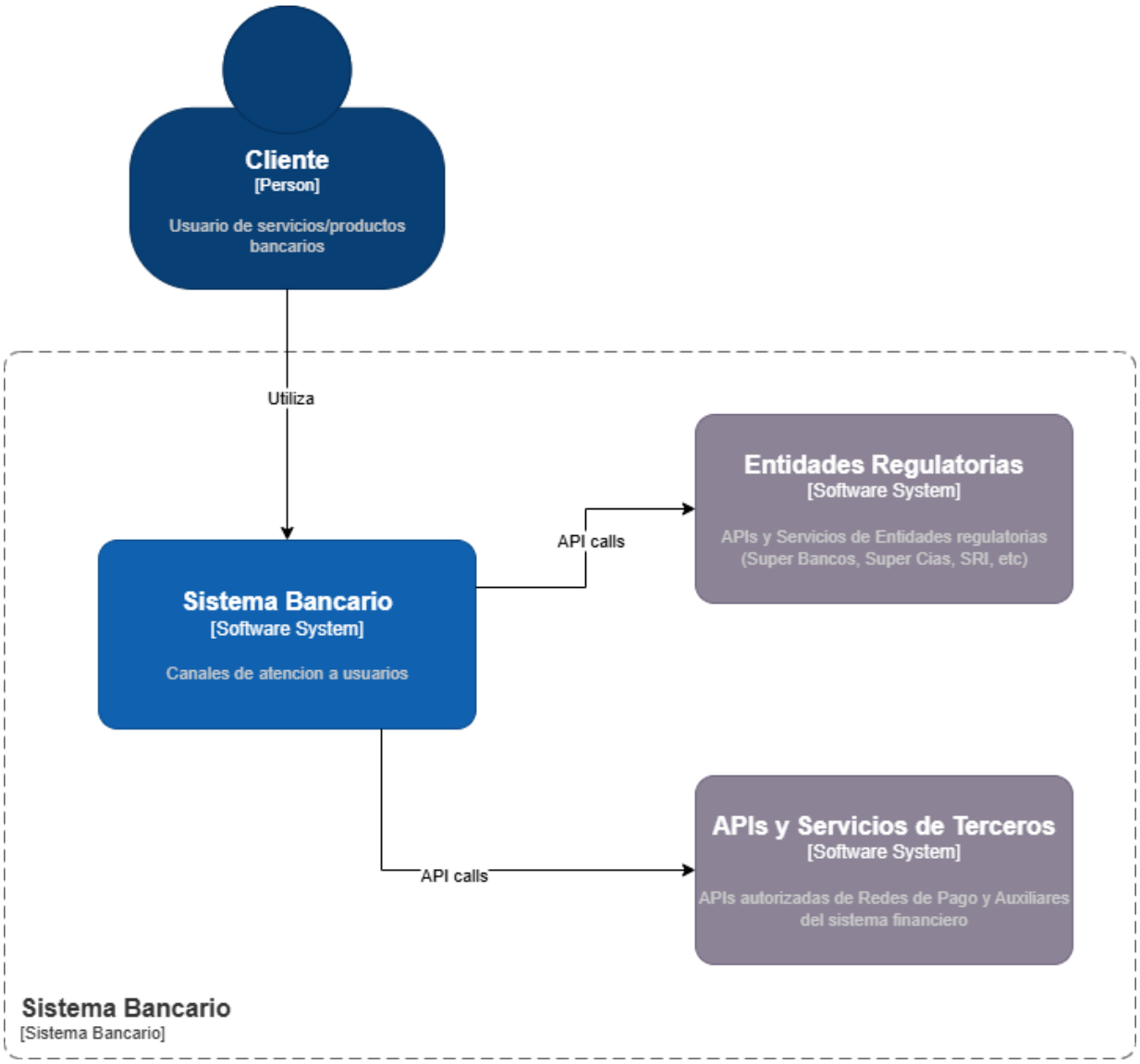
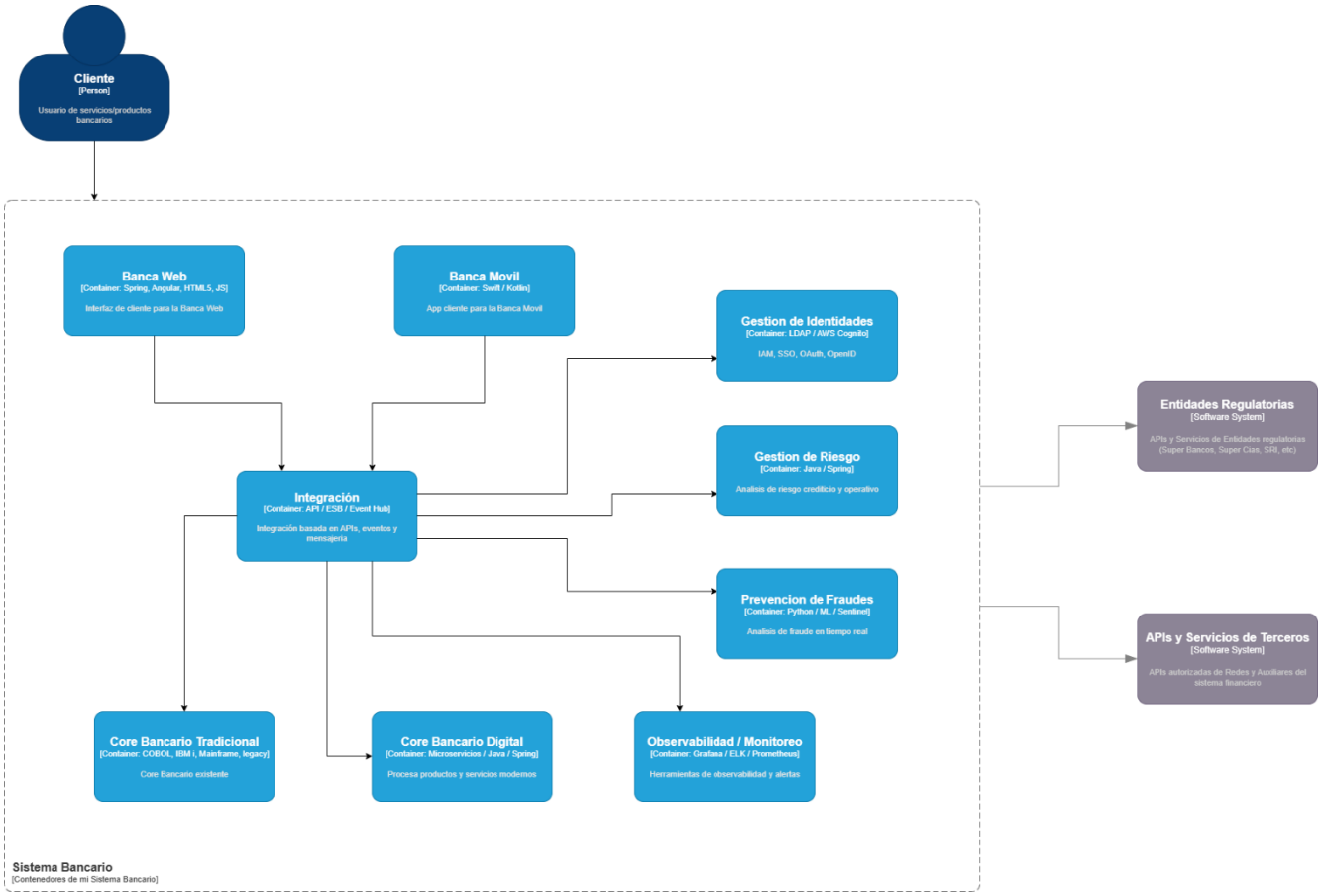


Diagrama de Contenedores:

En este nivel se puede apreciar los contenedores que formar el sistema en mención, aquí encontramos los siguientes:

- Banca Web: contenedor web de la banca en línea.
- Banca Móvil: contenedor de la app bancaria.
- Integración: componentes de integración que permiten la interacción de todos el sistema y sus dependencias.
- Gestión de Identidades: Gestión y control de identidades, RBAC, ABAC
- Gestión de Riesgo: Análisis del riesgo crediticio y operativo.
- Prevención de Fraudes: Análisis del fraude en tiempo real.
- Core Bancario Tradicional: Core legado ya existente.
- Core Bancario Digital: Core digital nuevo que procesa productos y servicios modernos.
- Observabilidad / Monitoreo: herramientas de Observabilidad, comportamiento y alertas.

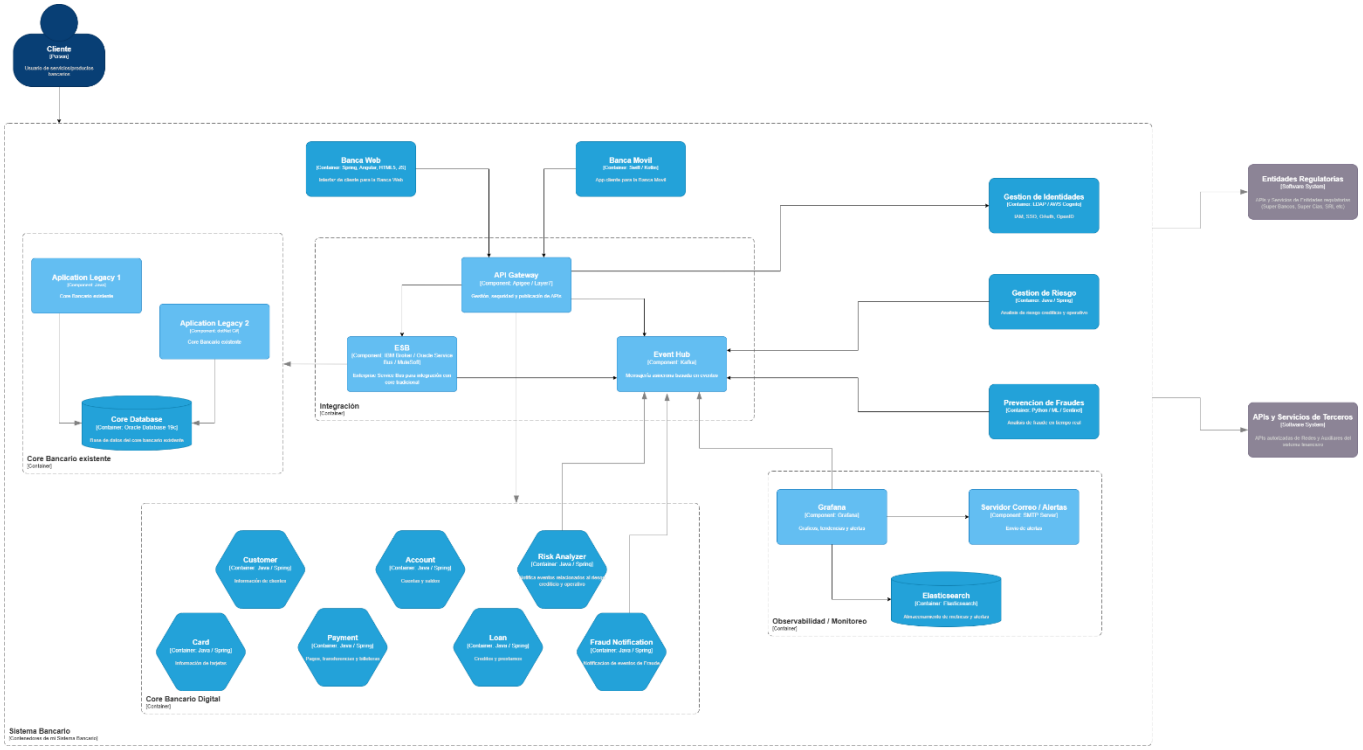


Dentro de este diagrama podemos observar que tanto la Banca Web como la Banca Móvil, que son los contenedores con los cuales interactúa el cliente, dependen directamente del componente de Integración, este componente basado en APIs, eventos y mensajerías, es el que va a orquestar los llamados hacia los demás componentes del sistema, ya sea al Core Bancario Digital o hacia el Core Bancario Tradicional existente. De igual manera se generarán los eventos necesarios para la

gestión de riesgos y la prevención de fraudes y todo lo que tiene que ver con autenticación y autorización, RBAC y ABAC, es gestionado por el componente de Identidades.

Entonces, el componente de integración orquesta todos estos llamados hacia estos componentes, y también entrega las métricas del funcionamiento de cada uno de ellos hacia el componente de observabilidad y monitoreo, el cual se encarga de analizar este comportamiento, estas métricas, estos umbrales, y producir las alertas respectivas del sistema. Aquí estamos usando un enfoque guiado por APIs (API-led connectivity), esto me permite modularizar las integraciones necesarias en el sistema estructurando las llamadas a los sistemas de acuerdo con su función y dominio.

Diagrama de Componentes:



En este diagrama identificamos el componente de integración que es el corazón de esta arquitectura, es el punto de convergencia entre el core legacy y el core digital. Aquí vamos a aplicar un patrón llamado Strangler Fig (higo estrangulado) que me va a permitir de alguna manera a través del API Gateway orquestar las llamadas que haga el usuario desde los canales ya sea web o móvil y decidir qué llamadas se hacen hacia el core legacy y qué llamadas se hacen hacia el core digital.

Con esta implementación yo aseguro que hay una transición, que puede haber un co-living de ambos core bancarios y lo que debería pasar es que poco a poco las funcionalidades del core legacy se deben ir migrando hacia el core digital, se deben ir implementando en un componente digital nuevo basado en microservicios como se puede ver en el gráfico de componentes.

Este core digital va a ir absorbiendo esas funcionalidades que tiene el core legacy y estableciendo un estándar alineado a BIAN, en donde tenemos varios dominios mandatorios en esta arquitectura de referencia, como el de Accounts, Loans, Payments, Customer, Card, etc, que me permitirán el intercambio con otras entidades financieras a través de este estándar abierto.

El componente de integración tiene básicamente un API gateway que es el que expone los servicios internos y externos como API REST, aplica las políticas de seguridad, monetización para la integración con terceros, para las pasarelas de pago, redes y auxiliares del sistema financiero, también hace de proxy entre los entes regulatorios y nuestro core.

Las tecnologías podemos utilizar para esto, puede ser Azure API Management, un AWS API Gateway, un Layer7 API Management de Broadcom, entonces entre las funciones clave que tiene este componente es la de autenticación, autorización, ya sea OAuth2, OpenID, transformación de mensajes, de SOAP a REST y viceversa para la integración con sistemas legacy, genera métricas y auditoría.

También nos encontramos con el componente Event Hub, que es un componente de una arquitectura basada en eventos que nos garantiza comunicación asíncrona y desacoplada, entonces, a través de la publicación y suscripción de eventos de negocio pueden identificarse lo siguiente: transacciones completadas, transacciones con error, alertas de fraude, o incluso actualización de score crediticio o de riesgo operativo en alguna transacción o canal.

Las tecnologías que se pueden utilizar para este componente pueden ser Apache Kafka, RabbitMQ, Azure EventHub o AWS EventBridge. Aquí aplicamos el patrón Event Sourcing, esto, a su vez, se integra con unos componentes de monitoreo y observabilidad como Grafana, como una base de datos de Elasticsearch, y un servidor de correo para enviar alertas.

Por último, tenemos un Bus de Servicios Empresarial (ESB), aquí las tecnologías que podemos usar son un IBM Broker, un Oracle Service Bus, MuleSoft. Lo que me permite este componente es básicamente ser un traductor, adaptar los servicios o interfaces legacy del core bancario existente a llamados o APIs publicadas en el API Gateway permitiéndonos decidir qué transacciones van hacia el core tradicional y cuales hacia el core digital. Con esta estrategia dual se mantendrá la transición gradual asegurando la operativa.

Infraestructura y Alta Disponibilidad:

Para el apartado de infraestructura y alta disponibilidad, se propone el despliegue en varias nubes y varias regiones. Podemos utilizar AWS y Azure como recomendación, para resiliencia y alta disponibilidad, si es necesario integrar con sistemas On-Premise se recomienda el uso de ExpressRoute de Azure o AWS Direct Connect para baja latencia.

También el uso de Kubernetes (EKS o AKS) dependiendo del proveedor de nube, Service Mesh (Istio) así tendremos balanceo y una infraestructura elástica en nuestro core digital y en la banca web y banca móvil.

Se debe asegurar toda la infraestructura a través de mutual TLS (mTLS) para que los componentes puedan asegurar su canal de comunicación. Se propone el uso de un Storage replicado para en varias zonas y la observabilidad con Grafana, ELK, etc.