SOLUCION AL EJERICIO PRACTICO DEV

Aplicación para arquitecto de Soluciones

DESCRIPCION DEL PROBLEMA

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, movimientos, productos y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle.

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).

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.

El sistema utiliza una 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.

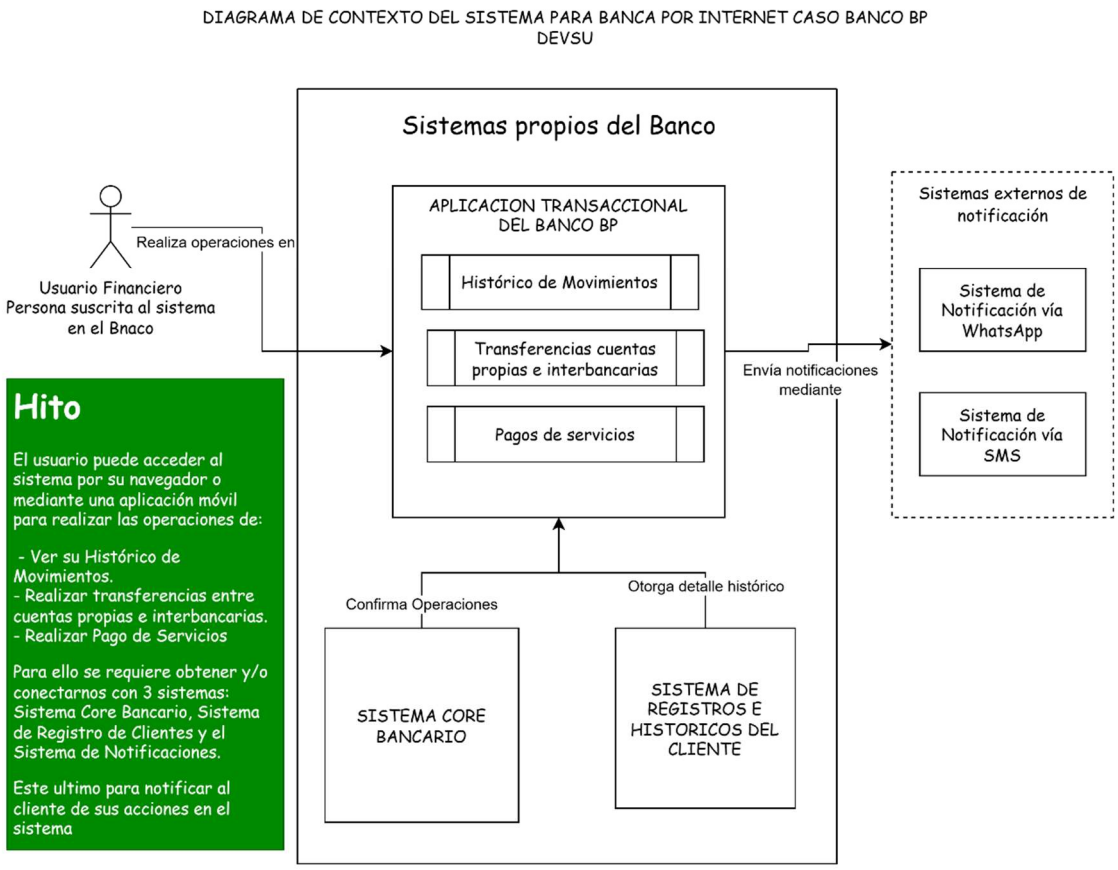
Para obtener los datos del cliente el sistema pasa por una capa de integración compuesta por un api Gateway y consume los servicios necesarios de acuerdo con el tipo de transacción, inicialmente usted cuenta con 3 servicios principales, consulta de datos básicos, consulta de movimientos y transferencias que realiza llamados a servicios externos dependiendo del tipo, si considera que debería agregar más servicios para mejorar el rendimiento de su arquitectura, es libre de hacerlo.

SOLUCION PROPUESTA

Para la solución inicialmente se proporcionará los primeros 3 diagramas solicitados, los mismos esta basados en la metodología C4.

DIAGRAMA DE CONTEXTO

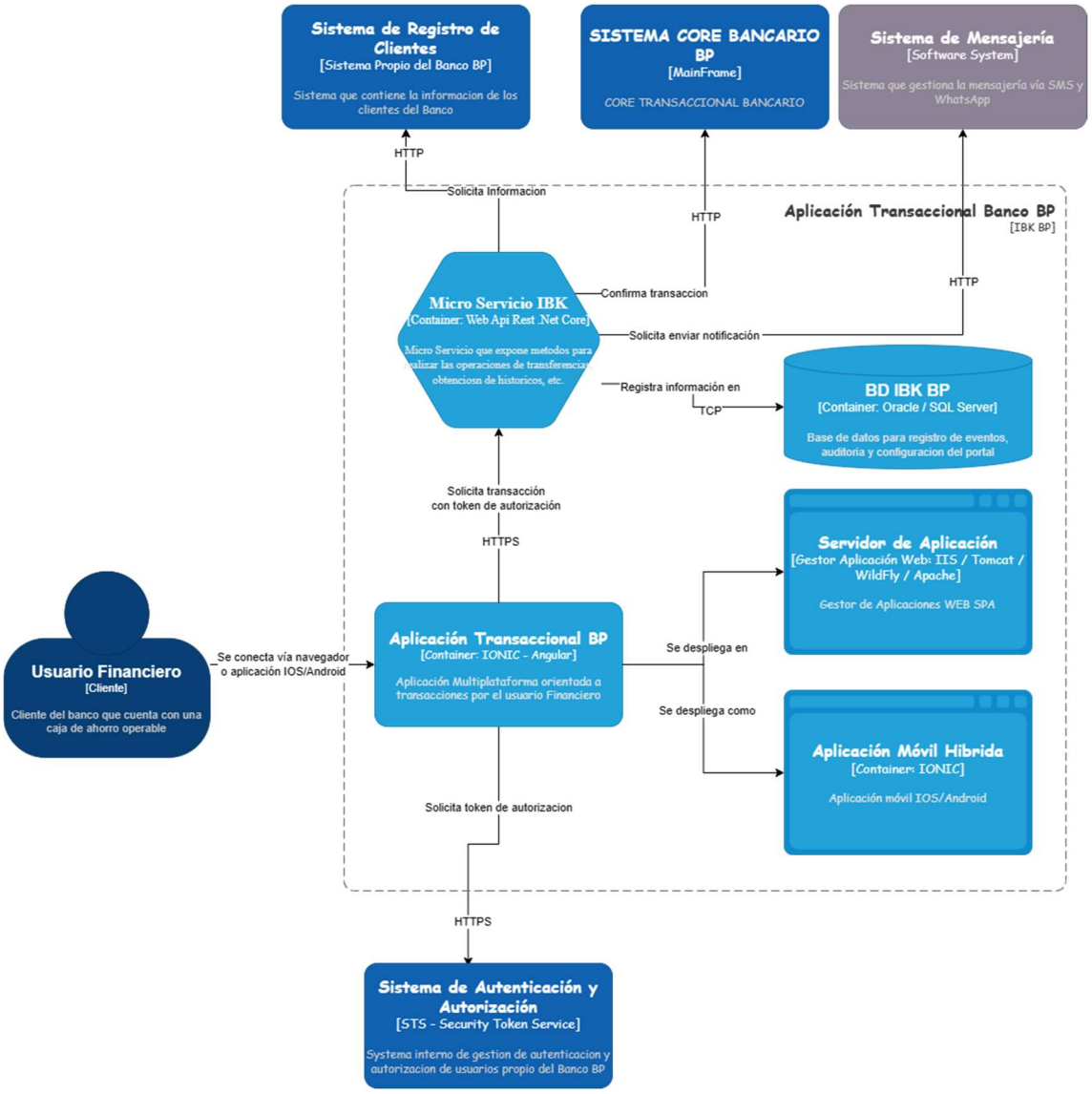
El mismo debe ser entendible por cualquier por cualquier stakeholder del proyecto y debe ser totalmente agnóstico a alguna base tecnológica.



En el anterior diagrama se muestra a gran escala los elementos del sistema y sus relaciones con otros elementos externos como son los usuarios y otros grandes sistemas. Adicionalmente se coloca el Hito como una historia de usuario, como mención para notar que cada hito pueda ir asociado a un diagrama de contexto.

DIAGRAMA DE CONTENEDORES

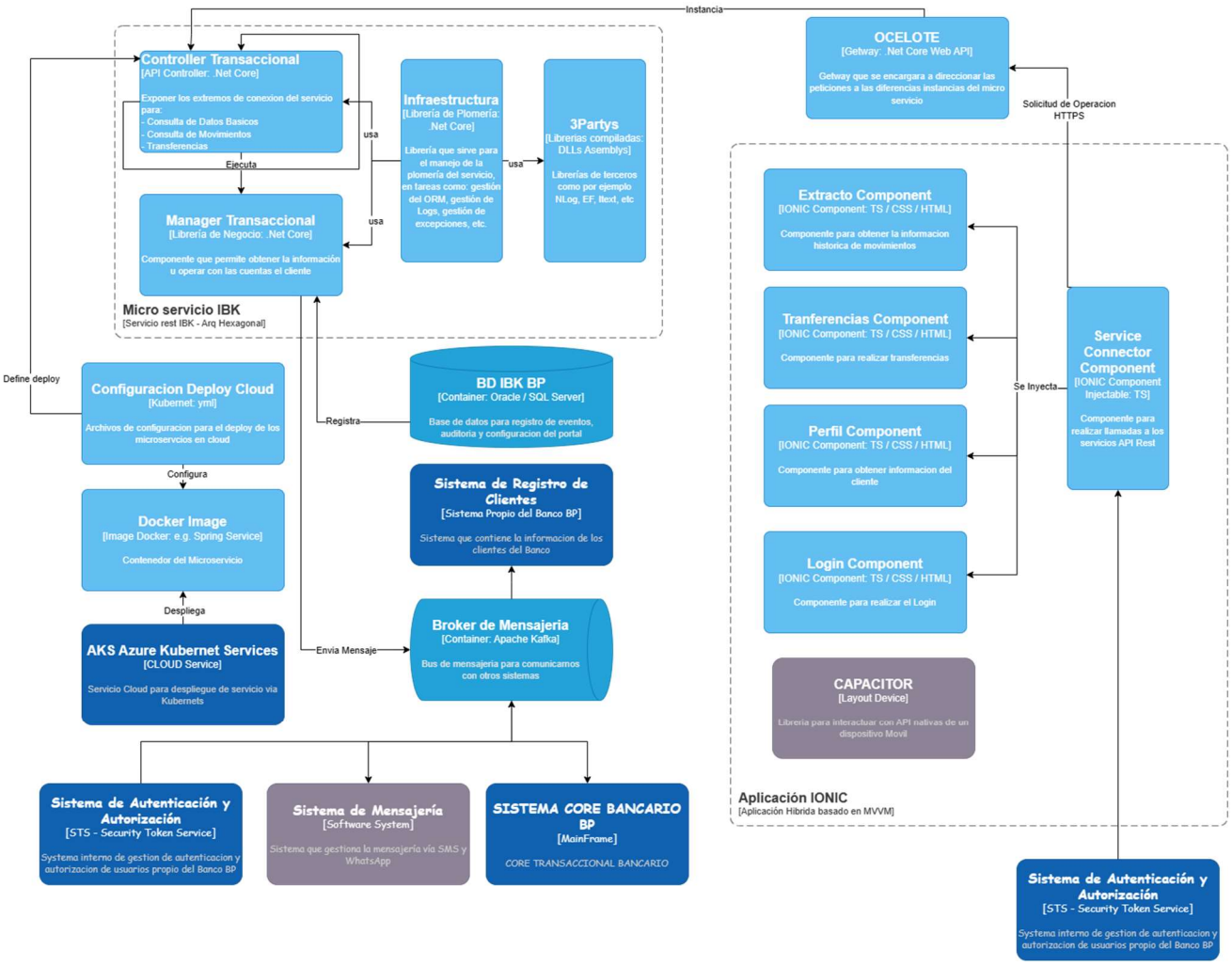
Este diagrama debe mostrar el tipo de componentes a usar con una tentativa de tecnología a nivel macro, por ejemplo un portal web, servicios soap/rest, etc.



El corazón del sistema propuesto se centra en los servicios que absorberán la funcionalidad transaccional orquestando la mensajería entre el cliente y los sistemas alternos y core del banco.

DIAGRAMA DE COMPONENTES

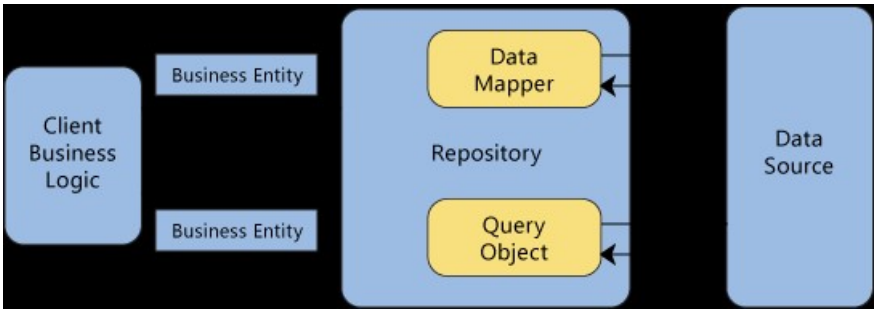
Este diagrama detalla los componentes necesarios para hacer factible la solución propuesta en el diagrama de contenedores.



En el anterior diagrama se describe la relación de los componentes necesarios para cumplir con los requerimientos funcionales, además de lograr las siguientes características en el sistema:

- Alta disponibilidad al tener la posibilidad de desplegar los servicios en una plataforma CLOUD, que a través de tecnología agnóstica como los kubernet's permitirá escalar el número de instancias en la medida que crezca el número de usuarios.
- Resilencia, ya que al pasar por un Getway como Ocelote se podrá instancias de los microservicios de mas de una instancia de los servidores en línea, en caso de alguna caída de uno de ellos, el resto asume la carga de forma equitativa.
- Monitoreo constante a través herramientas como Grafana al tener orquestado los mensajes entre el servicios central IBK con el resto mediante Kafkha.
- Autenticación y Autorización basado en OAuth 2.0, mediante el uso de un STS, mismo que permite federar la seguridad con otros sistemas de membresías (SAM) propios del banco, asi como con sistemas de autorización (SAA), que permite obtener un token al usuario posterior a una primera autenticación y después operar con el token en todas las interfaces de la aplicación, sin embargo se debe considerar los siguientes aspectos que están fuera del alcance del ejercicio:

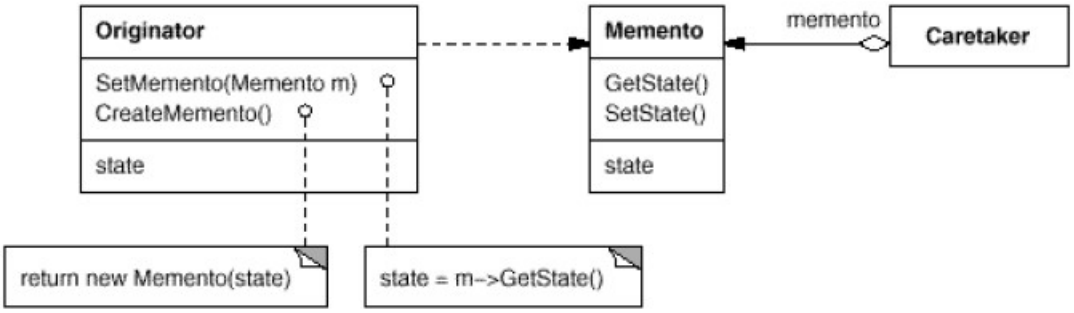
    - El uso de canales asociados a tokens, que permitan discriminar el acceso a micro servicios
    - Asociación de canales a los roles y perfiles de los usuarios.
- Separación de componentes 3Party, a través de una capa de infraestructura que permite escalar o reemplazar en las librerías de terceros, mediante el uso de inyección de dependencias a los manager's del negocio.
- Para el acceso a datos se usara la capa de infraestructura, mediante el uso del patrón repository asociado al contexto un ORM (se menciona Entity Framework en el diagrama):



DISEÑO DE SOLUCION PARA AUDITORIA

En el caso del auditaje de transacciones realizadas por la aplicación se debe considerar:

- Tener un registro de eventos por día (journal) en donde se detalle, lo siguiente por cada evento:
    - Detalle del request serializado enviado por el cliente
    - Token asociado a la transacción
    - Información de la sesión y membresía usada en la transacción.
    - Registro basado en fases, a partir de la llamada a algún servicio como el CORE, usar el patrón memento para guardar el ultimo estado del evento y continuar con la orquestación hasta finalizar el registro del evento en un flujo normal de transacción.
      Descripción del patrón memento:



- Realizar el registro de los eventos de forma asíncrona antes, durante y después de las llamas a otros servicios, para evitar detrimento el rendimiento de la aplicación.
- Usar la información para alimentar dashboards, datalakes y el uso de herramientas bigdata para encontrar patrones que permitan definir medidas de seguridad atómicas orientados a segmentos de clientes, que consideren montos máximos de transferencias, niveles de autorización o de demanda de uso de más factores de seguridad para ciertos tipos de operación.

CONSIDERACIONES NORMATIVAS – CASO BOLIVIA

Para la construcción de una aplicación transaccional vía internet, se recomienda lo siguiente (basado en la norma boliviana impuesta por la Autoridad de Supervisión Financiera ASFI):

- Tomar en cuenta en las historias de usuario los niveles de control de flujo de liquidez por cada cliente para evitar el lavador de dinero, en el caso de Bolivia bajo el registro de formularios PCC01.
- El diseño de voucher's y otros documentos que genere la aplicación por operaciones a terceros por parte del cliente, debe evitar mostrar información sensible como el saldo de la cuenta, bajo norma de secreto bancario.
- En caso de solicitud de información con antigüedad mayor a 3 años, la aplicación debe ser clara al respecto de no contar con la misma de forma instantánea, sin embargo, el banco debe tener la posibilidad de otorgar dicha información previa solicitud formal por parte del cliente, a coste de comisión bancaria amparado por al ASFI.
- En el caso de aplicaciones web, se debe considerar tiempos de reacción del sistema por inactividad normados por al ASFI, por ejemplo, a los 5 minutos de inactividad el sistema debe volver al logan de forma automática.
- Se debe tener un esquema de concientización a los usuario financieros para el uso conciente de los canales electrónicos para evitar caer en estafas electrónicas, por ejemplo el robo de identidad mediante phishing.
- Se debe tomar en cuenta las recomendaciones respecto a la gestión de las políticas de gestión de contraseñas por parte del banco, basado en la normativa ASFI, por ejemplo: no tener contraseñas planas en la BD, contar con un tiempo de expiración para las mismas, etc.
- Considerar el estándar NIS para la implementación de la aplicación.