Lenin Orihuela Ordoñez

Arquitectura de la Plataforma de Servicios Financieros

1.- Marco Legal :

Los servicios financieros que se ofertan en linea, desarrollan en base a criterios y normatividad de las autoridades como SBS (Superintendencia de Banca y Seguros) que promueve y controla la adopción de normativas y leyes que rigen dichas actividades como la Ley Nro 29733 de protección de datos personales. Permiten la adopción de servicios en nube de procesamiento de los datos de manera automatizada.

La adopción de un porcentaje de los procesos en la nube permite optimizar la calidad del servicio, permite reducir los costos operativos de la plataforma de servicios financieros. Así también permite agilizar los nuevos despliegues de “features” que necesite desarrollar mediante la implantación de automatización de tareas de testing, despliegue, desarrollo, devops.

2.-Funcionamiento :

2.1 Contexto :

La plataforma de Servicios financieros, tiene un grupo de micro servicios que estas corriendo en la nube en una VCP. Para todos los procesos que debe realizar necesita hacer integraciones :

- Con los servidores del Banco (On premise) donde se encuentra el Core Bancario, que guarda las operaciones, las bases de datos de los clientes, etc . Existe una conectividad en canal seguro hacia la interfaz que designa el banco para la comunicación con el micro servicio FACADE API Integration.
- La plataforma hace consultas a terceros como RENIEC (Base de Registro Nacional de Identidad)
- Una empresa que brinda el servicio de publicidad y avisos de las transacciones, ya sea por correo por mensajes de texto, .
- Una empresa que se encarga de recibir un contenido JSON donde se encuentran las características extraídas en la etapa de reconocimiento facial y forma parte del modulo “Onboarding”. El servicio de la empresa consistirá en validar la identidad de la persona asociada a las características que el modulo Onboarding envía en un formato JSON.
- Otras bases de datos externas, que consulta la plataforma para la validación de los perfiles de clientes .
- Así mismo de tiene un servicio para mensajería, encargado de generar vía email o mensajes de texto las operaciones que el usuario ejecuta dentro de la plataforma.

Dentro de la VPC, se encuentran :

a) Micro servicios de la plataforma que soporta la lógica del negocio
b) Solución de auto healing
c) Token de Acceso por el estandar OAuth2
d) Modulo de Monitoreo, el micro servicio con el nombre “MONITOR” interactuá como un orquestador, recibiendo los flujos de notificaciones, health check, Logs,métricas , usa una base de datos Dynamo , para la persistencia de la información
e) Los micro servicios están afectados por un balanceador de carga y también por auto scaling para el escalamiento horizontal automatizado, en caso la demanda se incremente la infraestructura incrementa el numero de containers para responder.
f) Para la parte de monitoreo hay un componente “MONITOR” que recibe los logs, las métricas, aviso de excepciones, health checking, para almacenar en una base de datos no relacional, los logs y datos que debo enviar a un servicio de visualización como KIBANA / trabajar con Prometheus y GRAFANA. Dentro de cada componente se ha habilitado patrones de monitoreo, tracking, health checking, esos flujos de datos son dirigidos al componente “MONITOR”, que los almacena, pero también sirve como alimentador de la solución “CodeDeploy”, para activar un despliegue generado por el aviso uno de estos indicadores-, de manera automatizada trata de restablecer el servicio. Este proceso se apoya en las soluciones de AWS.
h) La información Onpremise del Core Bancario serà obtenida por medio de una API Integration, componente del Sistema Bancario.
I) El sistema trabaja con una Cache distribuida, que almacena los Objetos (Con la información del cliente, en caso pase la validación de identidad. De esta forma los componentes pueden servirse de esta información, invocando al objeto para su consumo

2.2 Flujo de Trabajo :

El flujo de trabajo se da de la siguiente manera :

1.- Si la persona que consulta la pagina del banco por celular y no tiene una cuenta es llevado hacia el ingreso de sus datos personales, validación facial, donde el aplicativo del celular hace la extracción de características empaqueta esta información la enviá a la empresa externa con la que se hizo la integración, la empresa externa procesa la información y verifica la identidad asociada a ella, si cumple con las condiciones le responde al modulo ONBOARDING.

2.- Una vez que esta registrado como cliente donde ha creado un usuario y contraseña, ha firmado un contrato (mediante el reconocimiento fácil, la firma equivale a pasar por el reconocimiento facial) con ello la compañía externa hace esta validación y da conformidad (o firma el documento

si pasa correctamente el proceso) la empresa externa devuelve los documentos “firmados en pdf” para ser enviados al cliente por un correo electrónico.

3.- Una vez que el cliente ha creado su cuenta, tiene acceso al sistema por una contraseña y usuario , adicionalmente puede ser también por una huella digital. Puede ingresar al sistema con uno de los métodos indicados. Si la validación es exitosa el micro servicio “ CLIENTE ” que coordina con el componente "CONTROLLER" para interactuar con e MS API Gateway Integration y se comunica al Core Bancario, para traer la data del usuario en cuestión, Toda esa información, de cuenta, histórico, productos financieros se consolidan en un solo objeto o componente que debe ser almacenado en la Cache Hazelcast(Opcional) , de manera que los MS tienen la información que necesitan para sus flujos de trabajo del cliente que ha ingresado al sistema. Esta información que esta activa en la “Cache” puede permanecer activo 24 Horas.

4.- El usuario una vez logeado elige una operación a realizar . Ambos componentes “ Cliente” y “Controller" trabajan juntos para que por medio del componente “ API integration Gateway “ se comunique con el “Core Bancario”, para obtener los estados de cuenta, históricos, productos financieros, hacer un operación de deposito o retiro. El componente “API Integration” maneja los flujos operativos con los que trabajan los servidores Core del Banco. La extracción de la a información de acuerdo a la operación solicitada, se apoya en el sistema “ Cache”, como intermediario, de manera que los componentes se sirven de este almacenamiento.
Cuando el cliente es usuario frecuente de los servicios financieros, al ingresar en el mismo día varias veces, tiene menores tiempos de acceso ya que su información persiste 24 Horas(configurable) en el sistema Cachè (puede ser Hazelcast”) . Su información esta lista para ser consumida por los componentes.

5.-La respuesta a un Arquitectura de Alta disponibilidad, la abordamos con la solución de AWS de Auto Scaling, que permite como se mencionó el crecimiento horizontal de containers y Computo EC2 de manera dinámica, cuando no es necesario el recurso se deja de usar y de esta manera no se incrementan los costos del servicio, solamente funciona cuando se necesita. Adicional a esto se considera un Balanceador de Carga también.

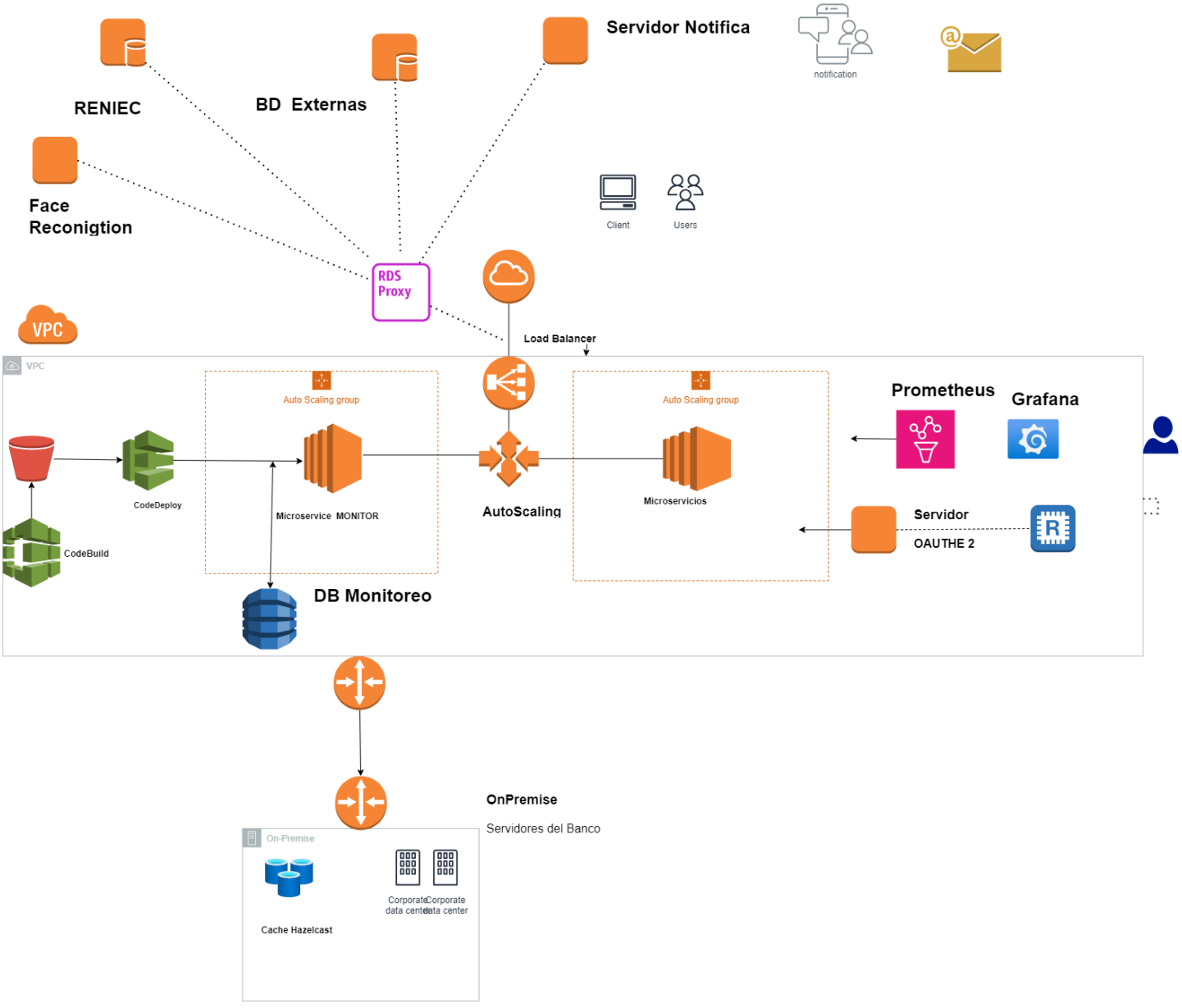
6.- La parte de Auto Healing, se plantea usar la solución CodeDeploy para el despliegue automático con sincronización por medio del componente “MONITOR” que envía la señal de activación a un “ chekpoint”.

7.- Para la etapa de “Logeo”, se considera un Servidor de Tokens de Acceso, que trabajan bajo el estandar Oauth 2.0 se usa un almacenamiento tipo cache REDIS de(equivalente en AWS).

8.- Cuando el cliente hace cualquier operación esta siendo registrada y almacenada en una base de datos no relacional (Dynamo),pero cuando el cliente necesita firmar un contrato de préstamo u otro producto financiero, se necesita el servicio de una empresa externa que integre y valide la identidad del cliente por reconocimiento fácil. Una vez que pasó la validación la misma empresa firma en cada contrato y genera un pdf que regresa al componente “CLIENTE” que posteriormente

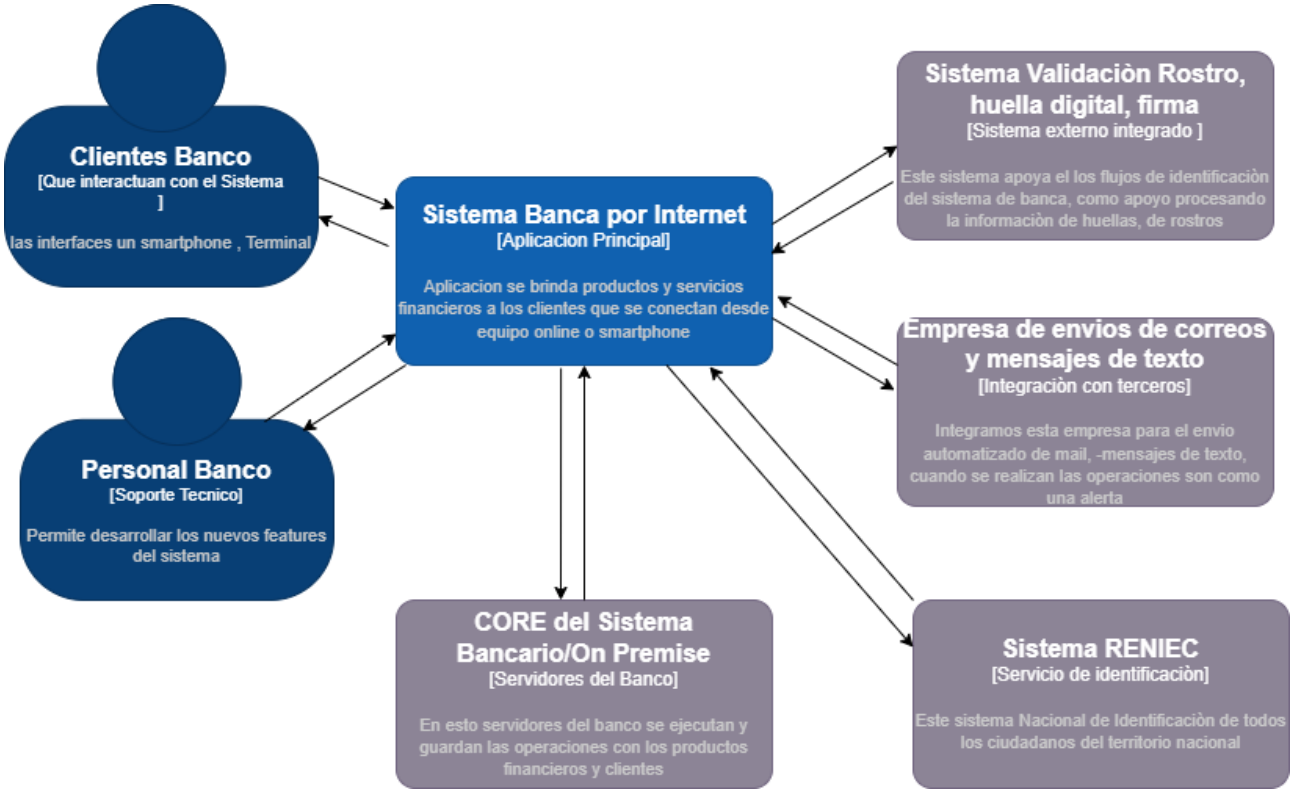
enviá al componente " MENSAJERIA", donde se envía la documentación firmada a un correo electrónico. De la misma manera cuando se realizan las operaciones del cliente, estas son enviadas al componente mensajería para que por email o mensaje de texto haga llegar al cliente.

3.-Modelo de Infraestructura

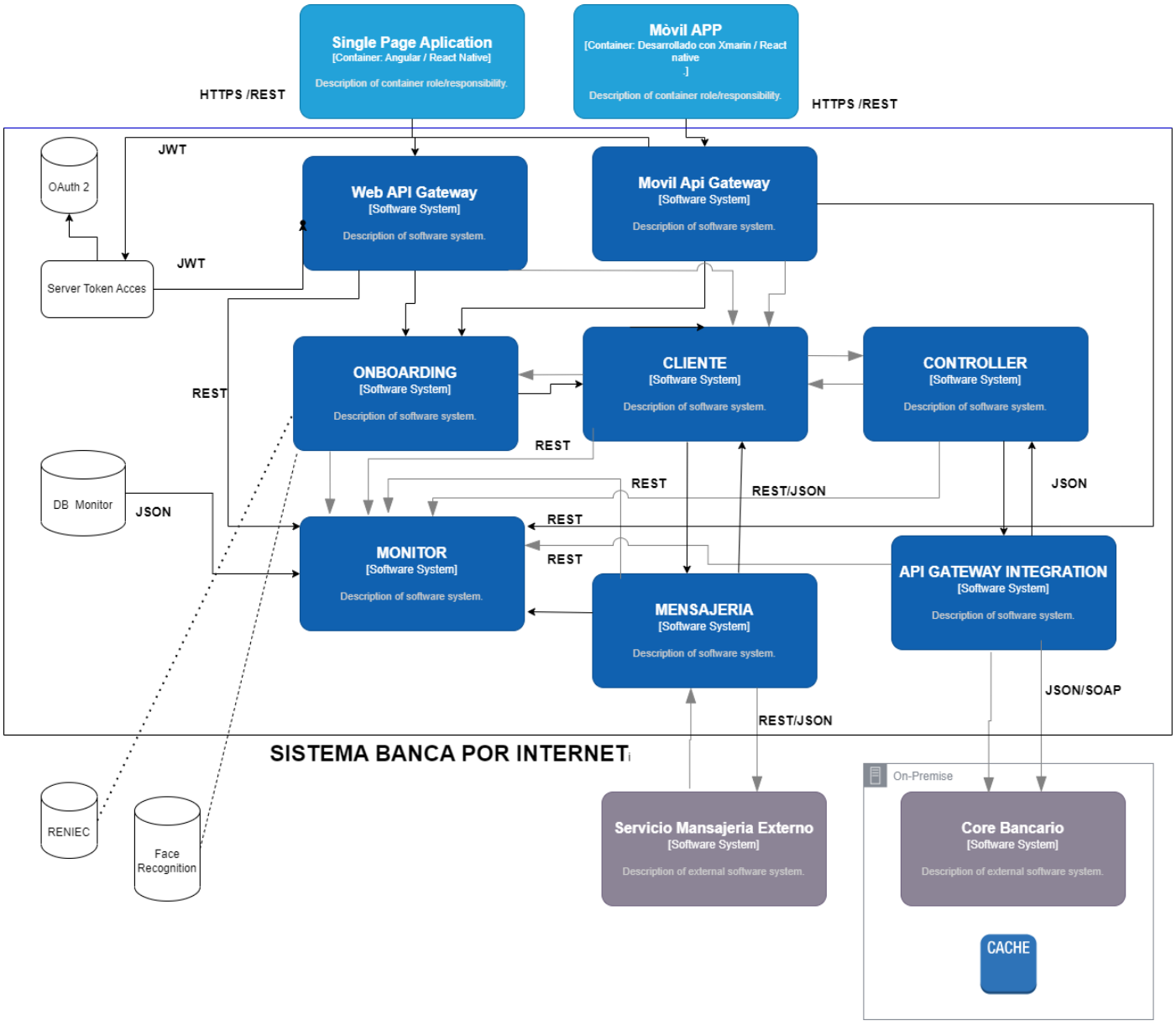


4. MODELO C4

4.1.- Modelo de Contexto

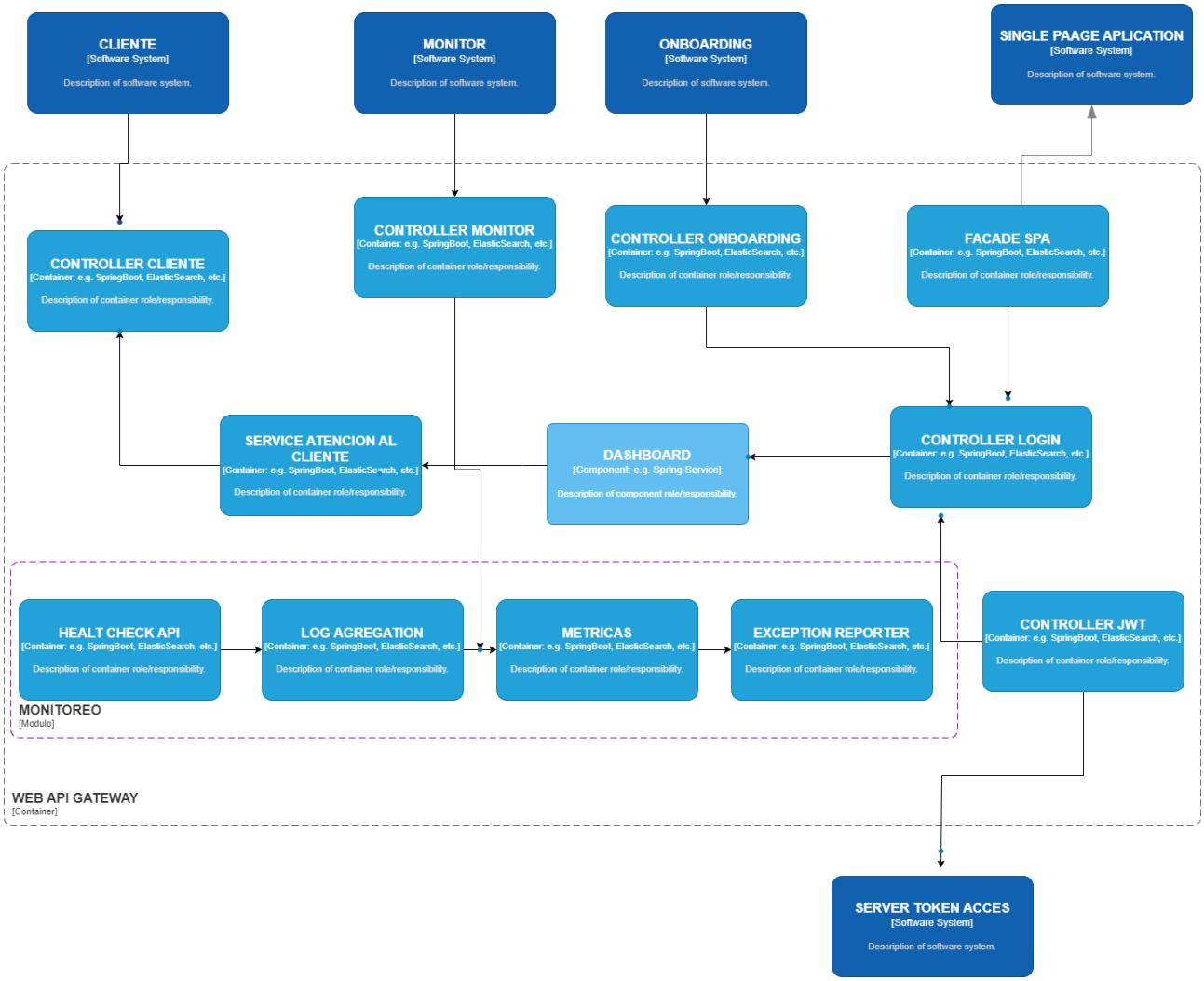


4.2.- Modelo de contenedores

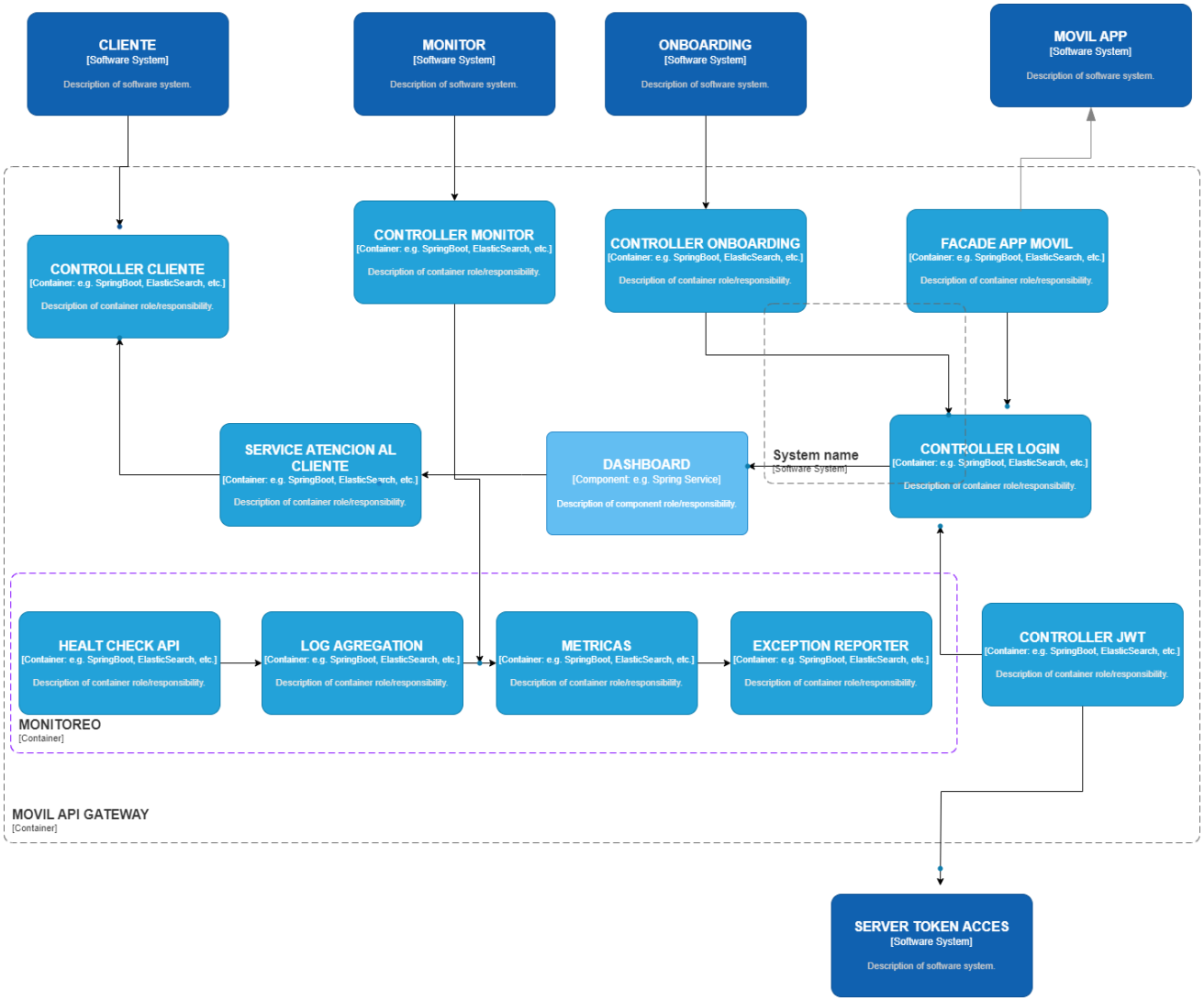


4.3 Modelo de Componentes

4.3.1 Componente Web API Gateway

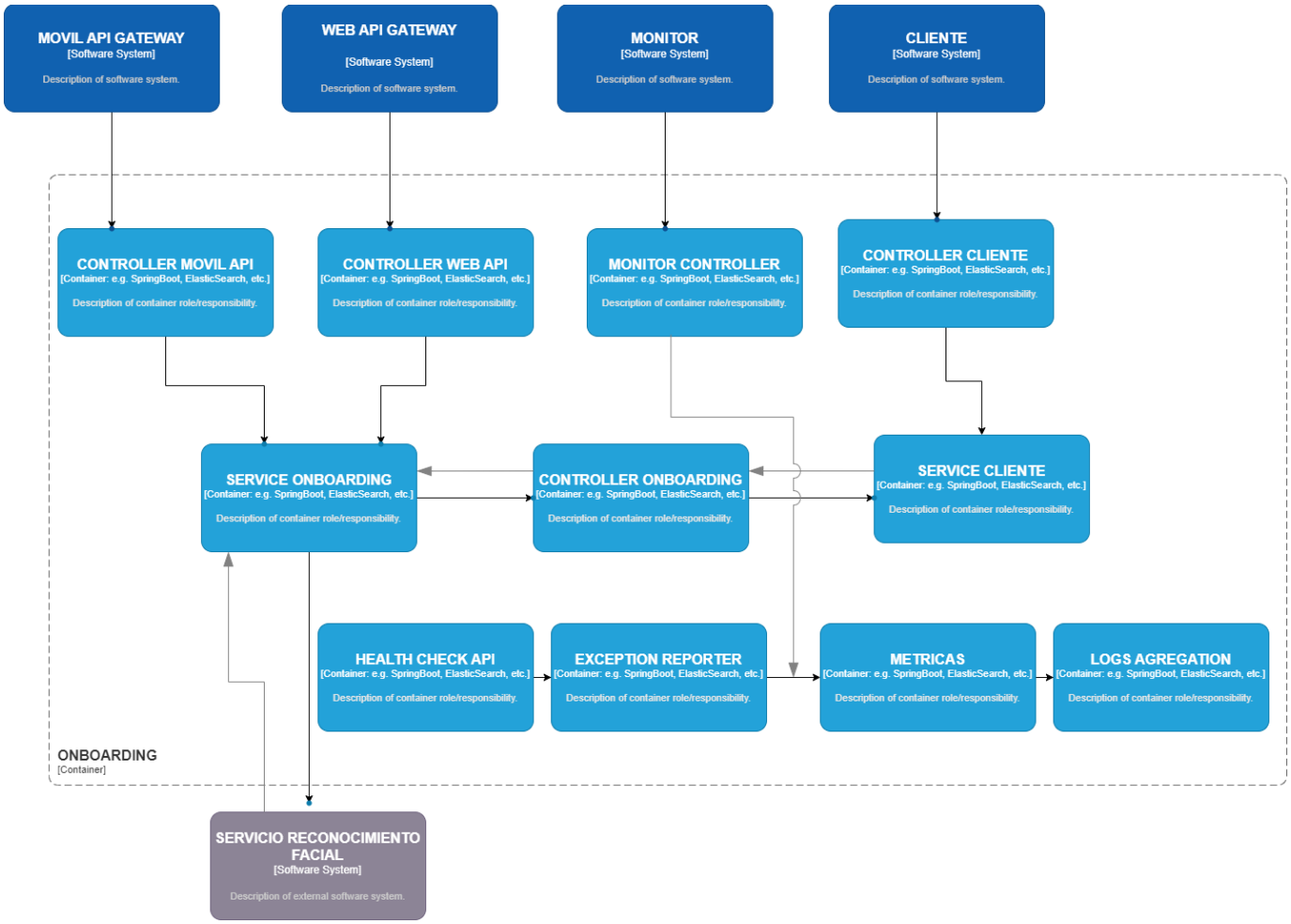


4.3.2 Componente Móvil API Gateway



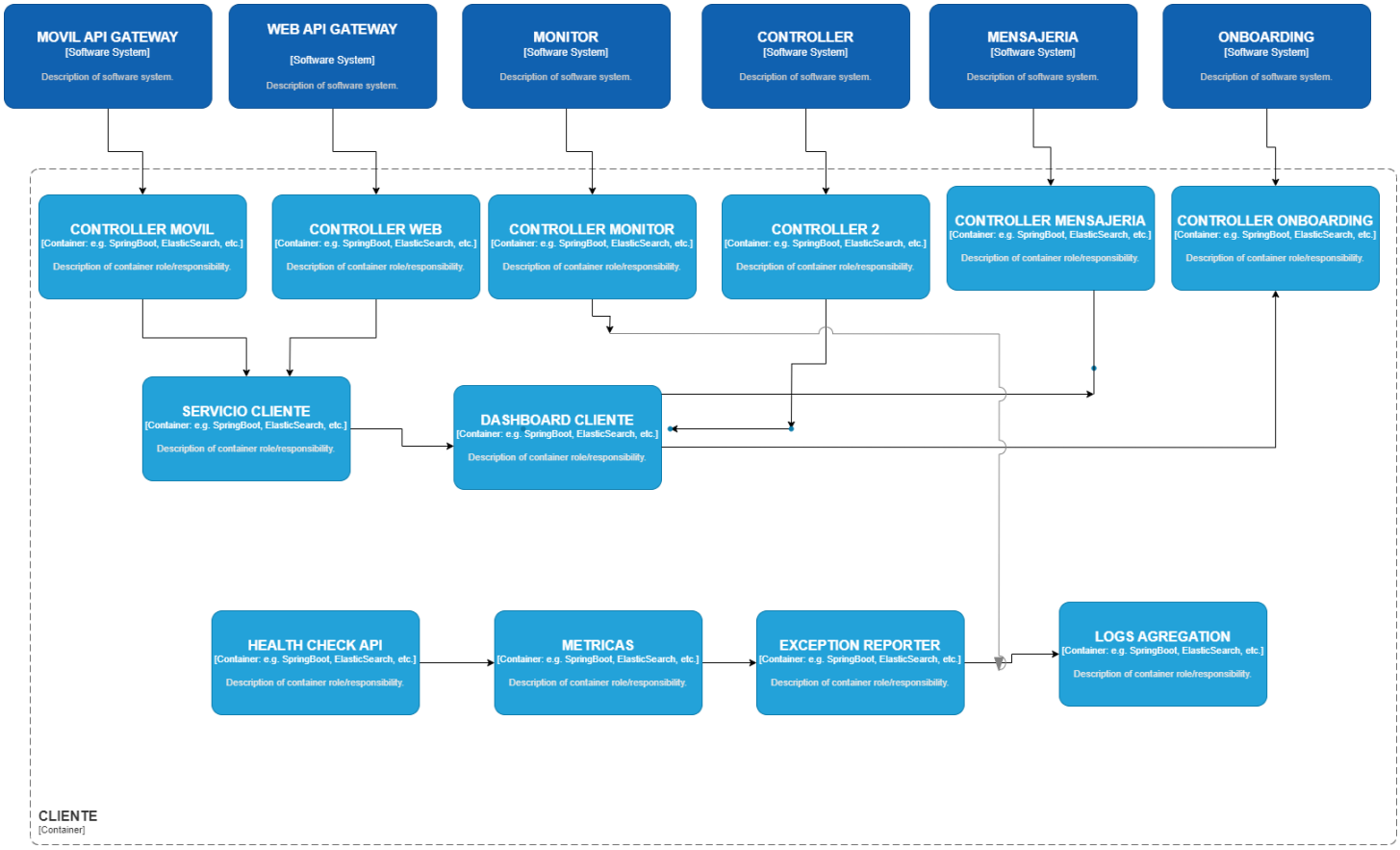
.

4.3.3 Componente Onboarding



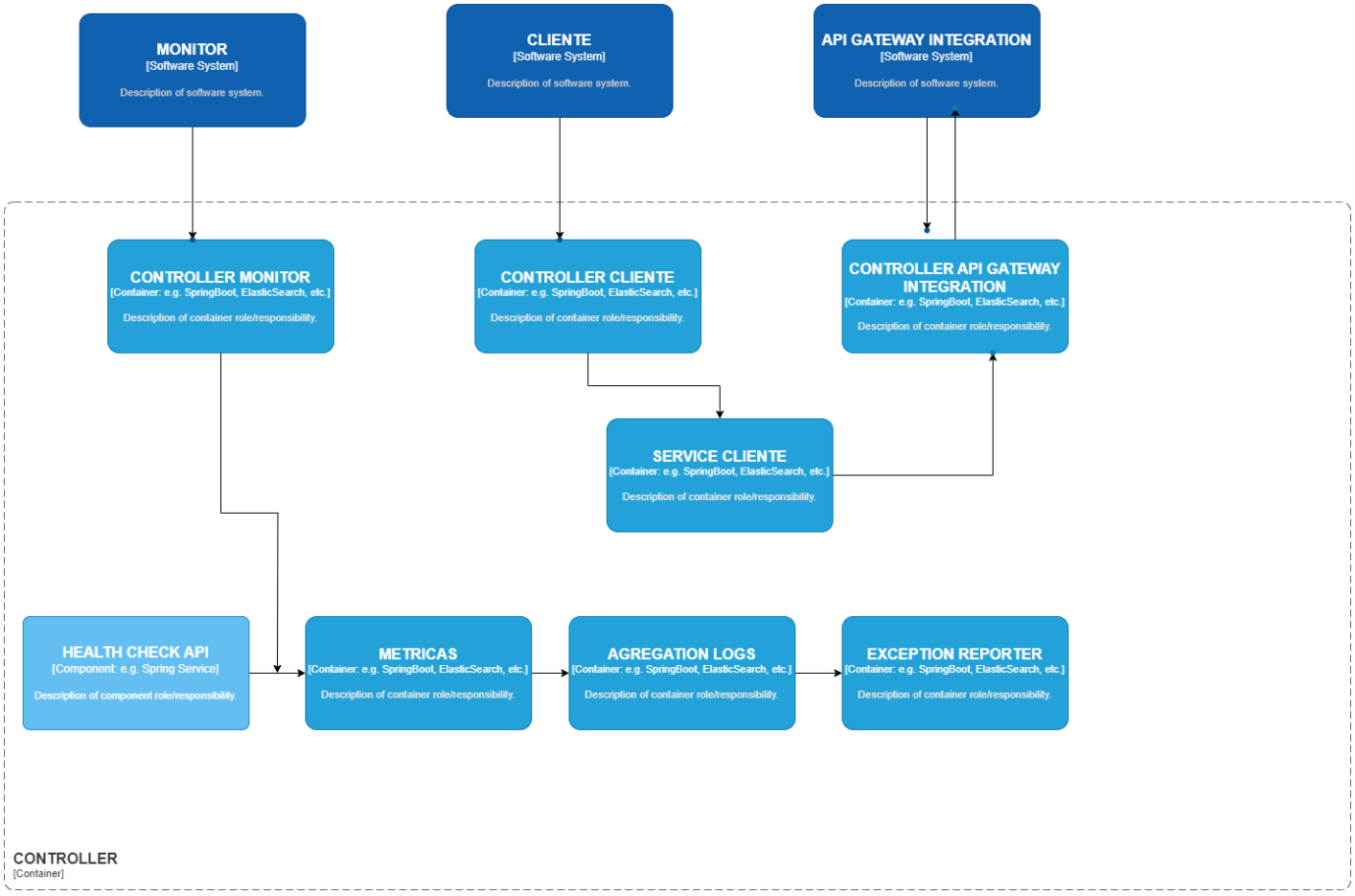
.

4.3.4 Componente CLIENTE



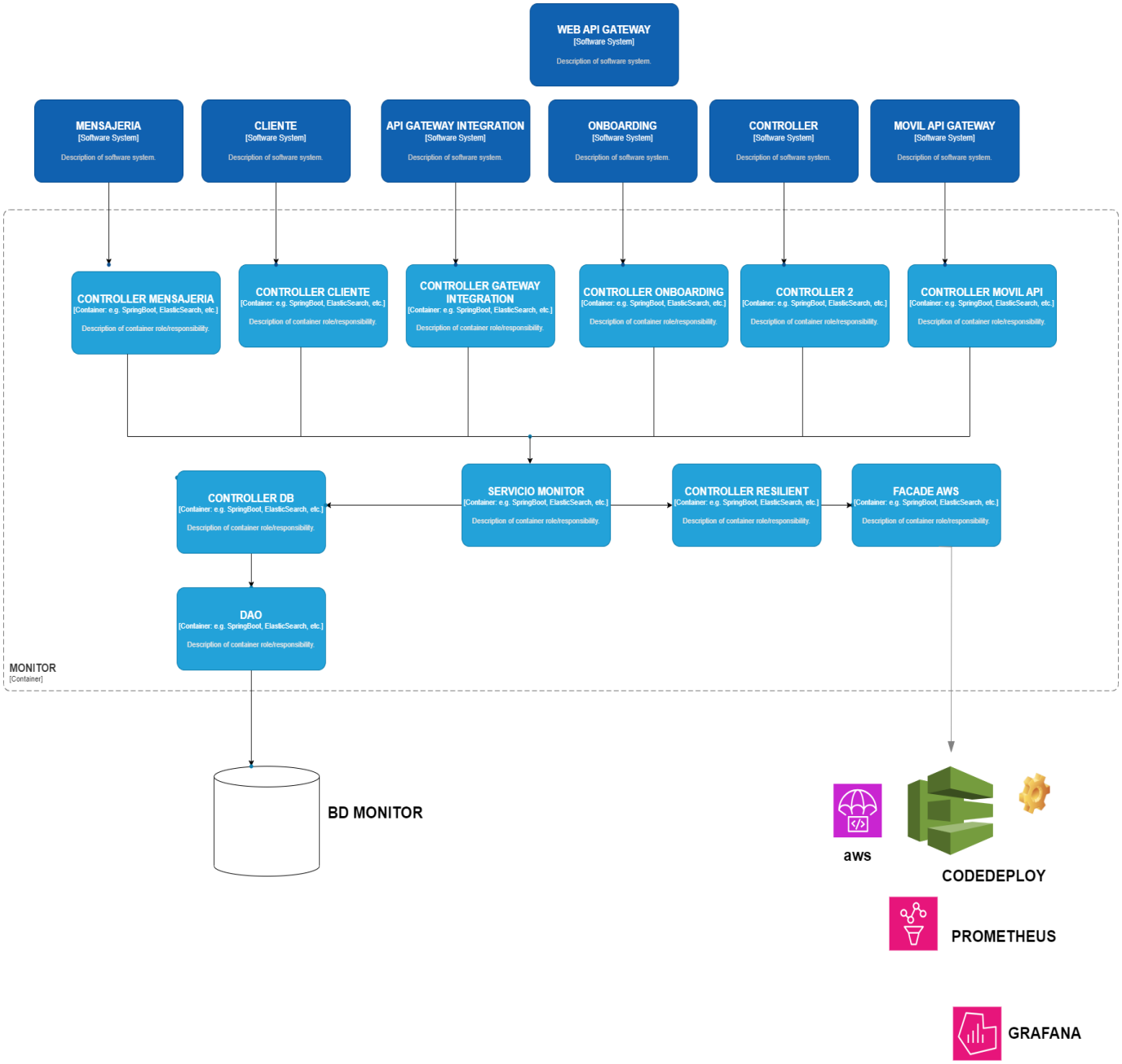
.

4.3.5 CONTROLLER



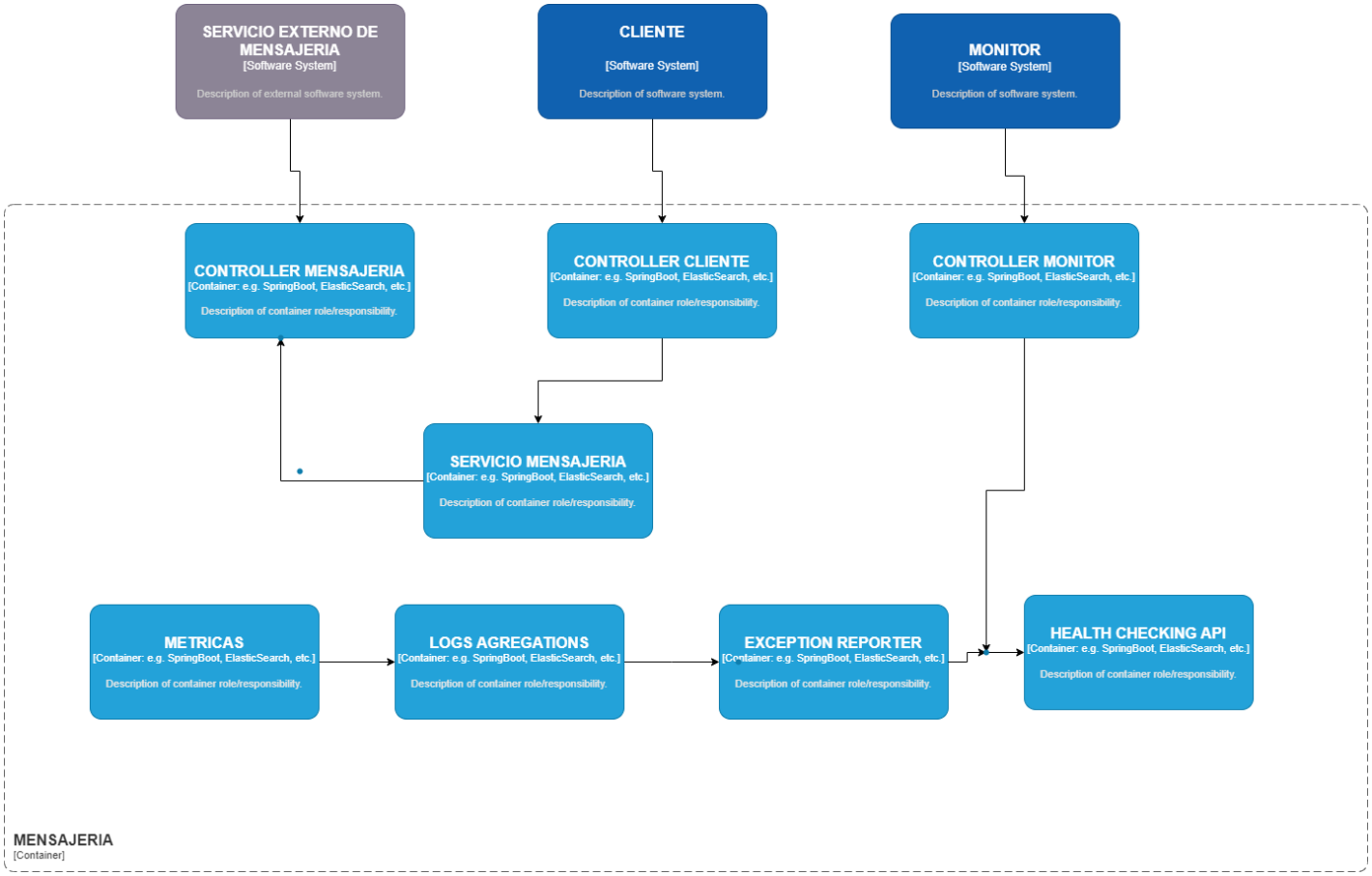
.

4.3.6 Componente MONITOR



.

4.3.7 MENSAJERIA



.

4.3.8 API GATEWAY INTEGRATION