

DEVSU

Correo electrónico: [correo electrónico aquí]

Sitio web: <https://devsu.com/>

DISEÑO DE INTEGRACION



Francisco Schnabel

Correo electrónico fasz58@hotmail.com

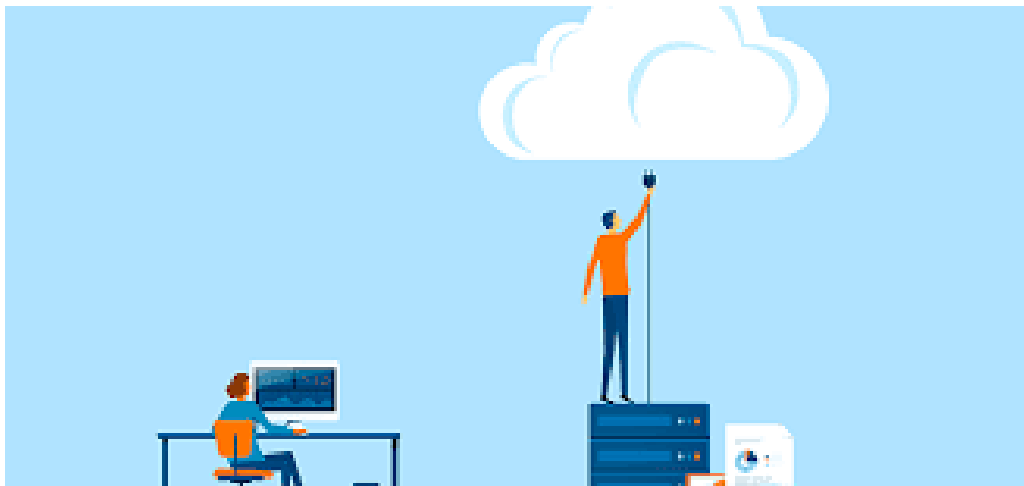
CONTENIDO

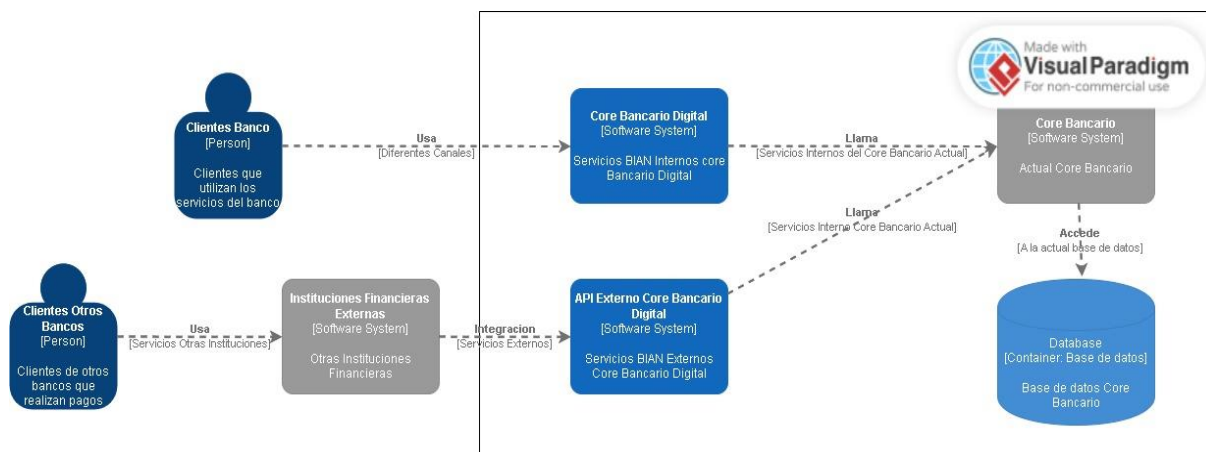
| | |
|---------------------------------------|---|
| Introducción | 1 |
| Contexto | 2 |
| Contenedor Cliente Banco | 3 |
| Contendor Cliente Otro Banco | 4 |
| Componentes Cliente Banco | 5 |
| Componentes Clientes Otro Banco | 6 |
| Recomendaciones | 7 |

Introducción

Para abordar el problema en cuestión, se ha diseñado una solución de integración que satisfaga las necesidades de migración de infraestructura hacia una modalidad en la nube.

Para ello, se ha utilizado el modelo C4 para presentar la solución. Es importante tener en cuenta que este diseño representa un punto de partida para el proyecto. A lo largo del desarrollo, es posible que se requieran ajustes para adaptarse a las necesidades cambiantes del cliente o a factores externos imprevistos



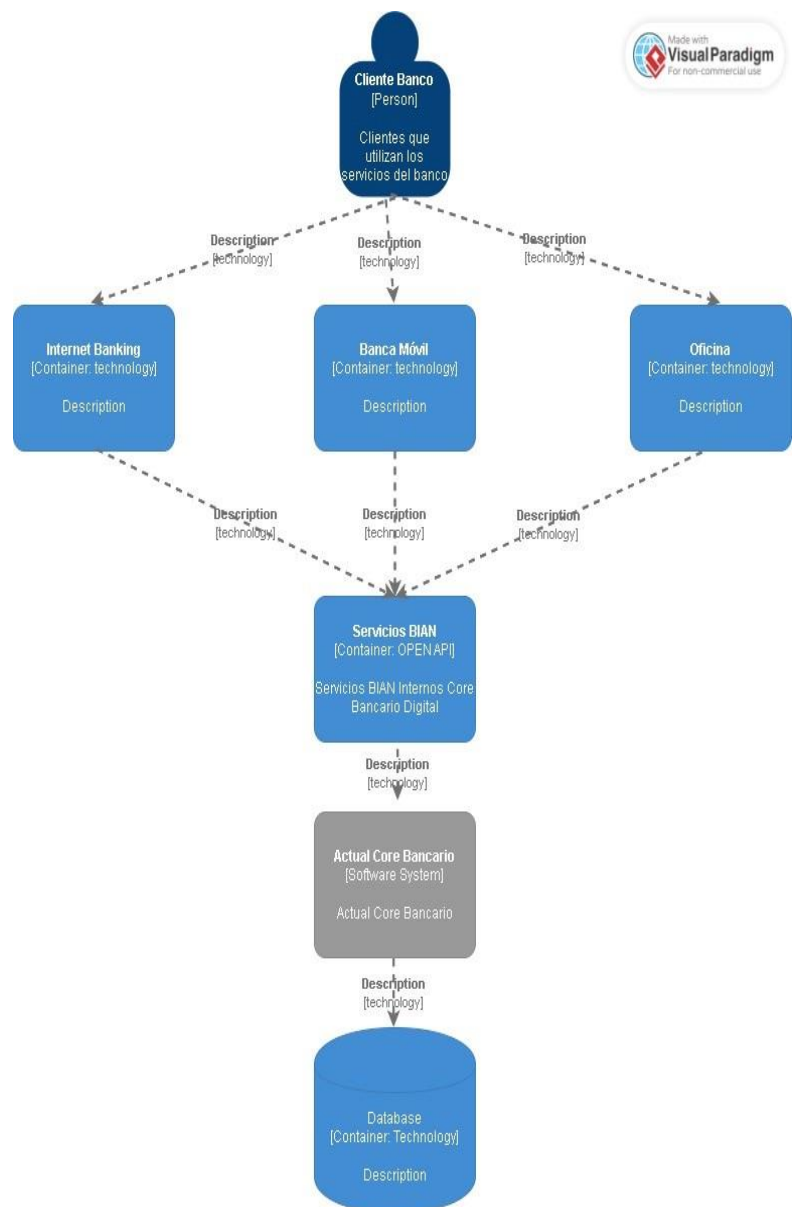


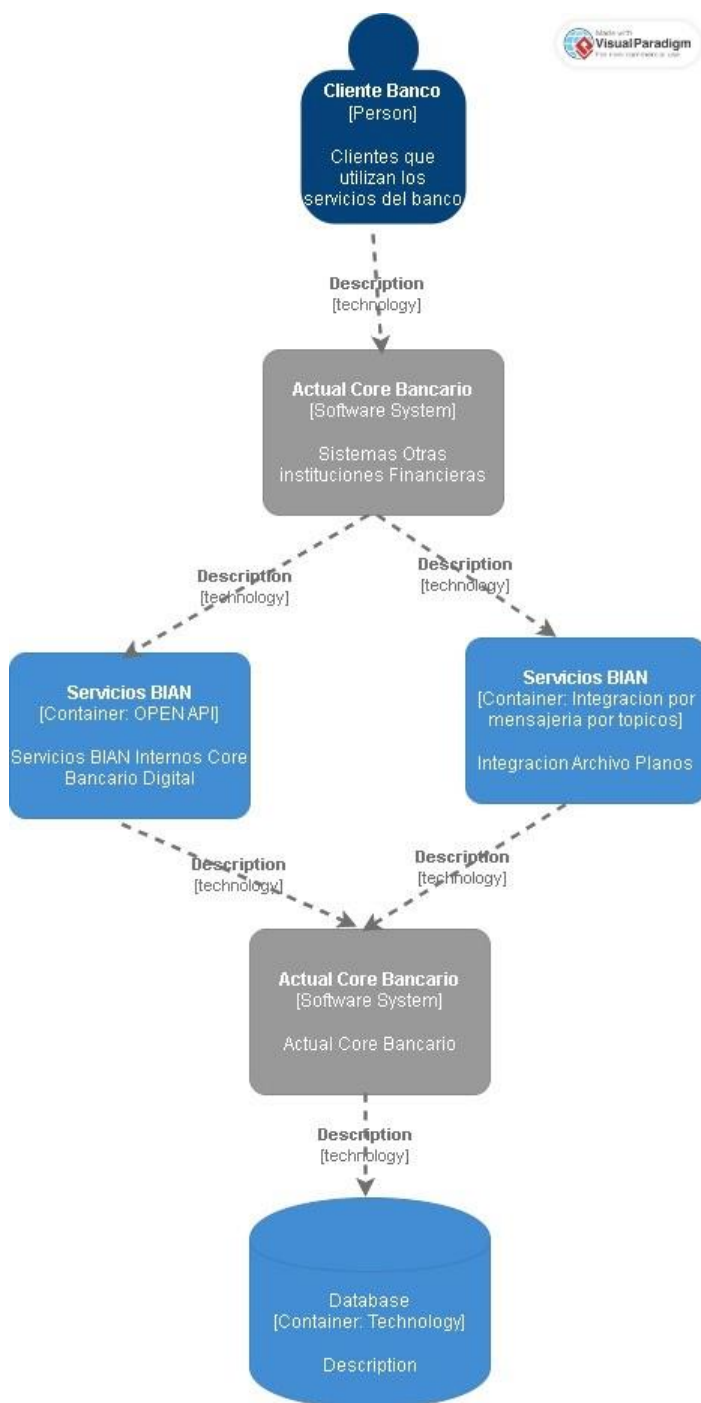
Contexto de la integración

La migración al núcleo bancario digital exige la definición de APIs internas y externas siguiendo el esquema BIAN. Este enfoque permite establecer un gobierno de servicios normalizados que garantice que las APIs cumplan con sus funciones específicas y no realicen tareas para las que no están diseñadas.

Clientes del Banco

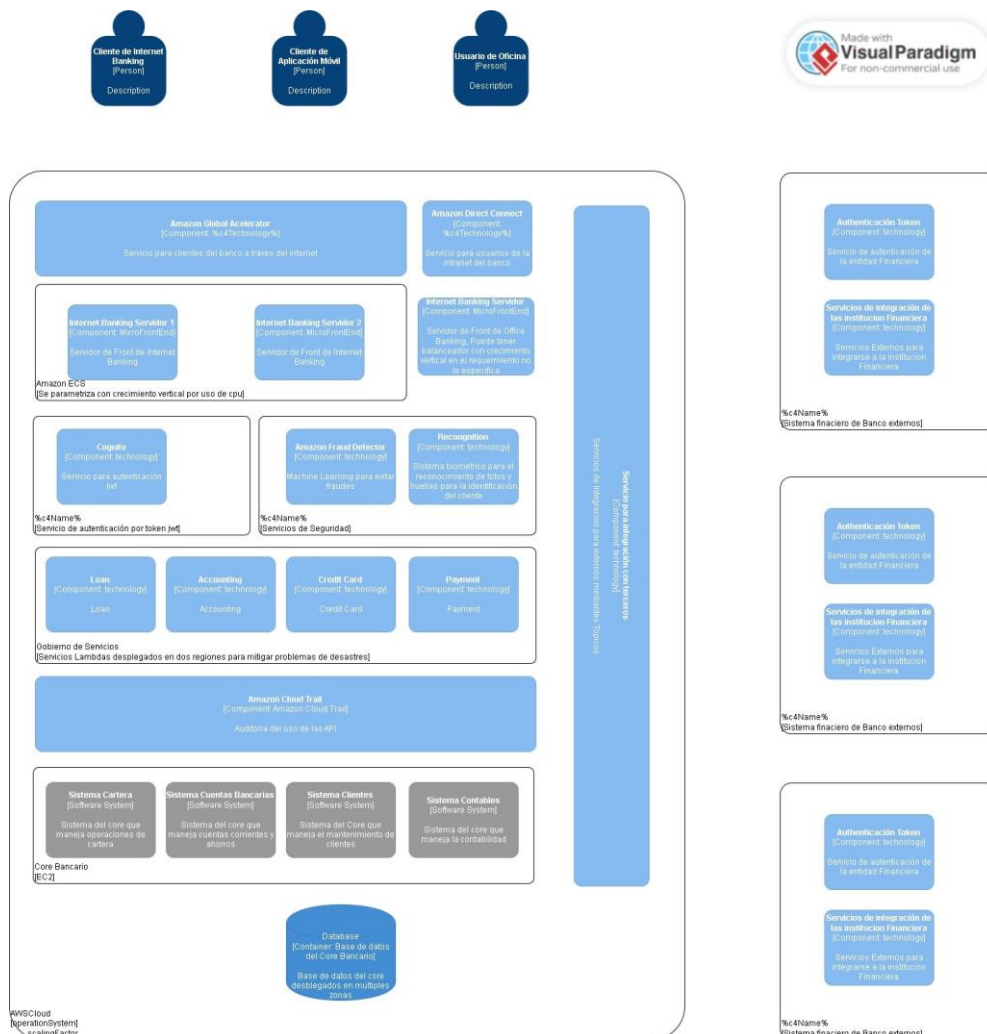
Los clientes del banco acceden a los servicios del núcleo bancario a través de Internet Banking, una aplicación móvil o visitando una oficina. Al utilizar estos sistemas, se conectarán a un dominio de servicio que inicialmente gestionará las solicitudes de la nueva banca digital. Este dominio de servicio será el intermediario entre los clientes y al bancario actual, que a su vez se conectará con la base de datos.





Clientes de otros Bancos

Clientes de otras instituciones financieras utilizan una variedad de servicios ofrecidos por diferentes bancos y, en ocasiones, necesitarán realizar transacciones con nuestro banco. Al operar con sistemas externos sobre los cuales no tenemos control, deberán utilizar las APIs expuestas para clientes externos. Estas APIs, a su vez, se comunicarán con los servicios del bancario actual, que se encargará de interactuar con la base de datos.



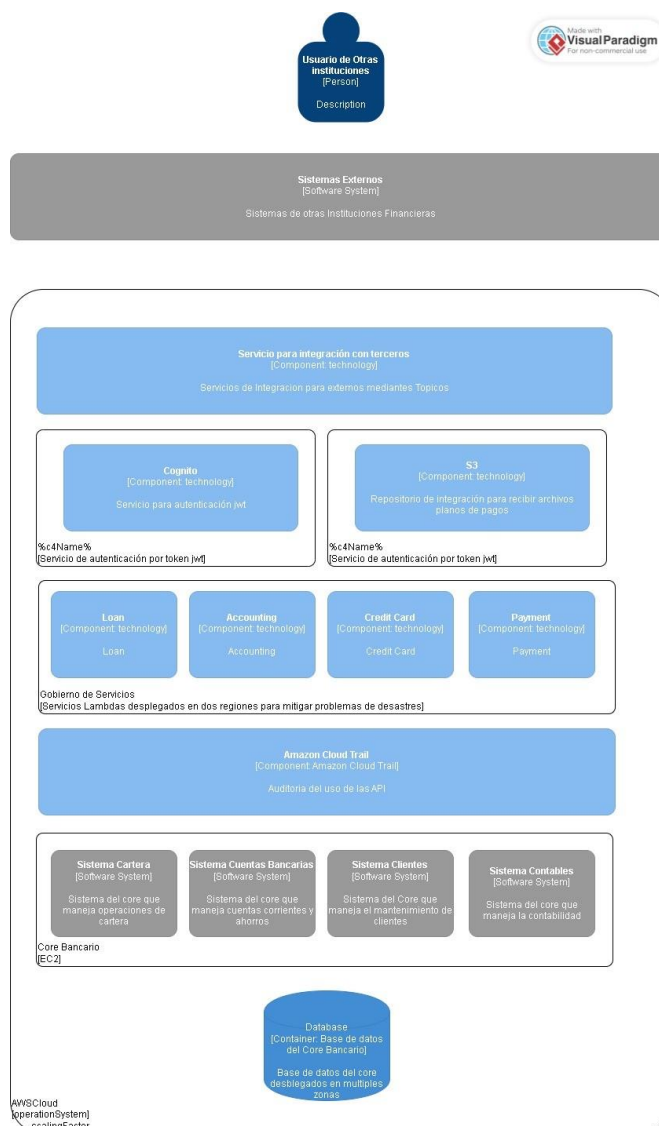
Componentes que utilizar con clientes del Banco

Para garantizar una baja latencia y mejorar la interacción de las aplicaciones web o móviles, se han considerado varias alternativas. En este caso, se ha optado por utilizar CloudFront para optimizar la entrega de contenido a través de Internet y Direct Connect para las oficinas, asegurando una conexión dedicada y de alta velocidad.

Además, se implementará una solución de machine learning para verificar que las transacciones sean realizadas por los clientes autorizados. Este sistema puede complementarse con software de reconocimiento biométrico para asegurar que la persona que realiza la transacción sea efectivamente el cliente legítimo.

Componentes que utilizar con clientes otros Banco

Debemos establecer múltiples puntos de integración con otros bancos, dependiendo de su origen o tecnología. Para ello, se dispone de una API financiera y un repositorio para archivos planos. En todos los casos, estos puntos de integración finalizan su proceso llamando a la API, la cual consume los servicios que el núcleo bancario proporciona para llevar a cabo la transacción solicitada.



Recomendaciones

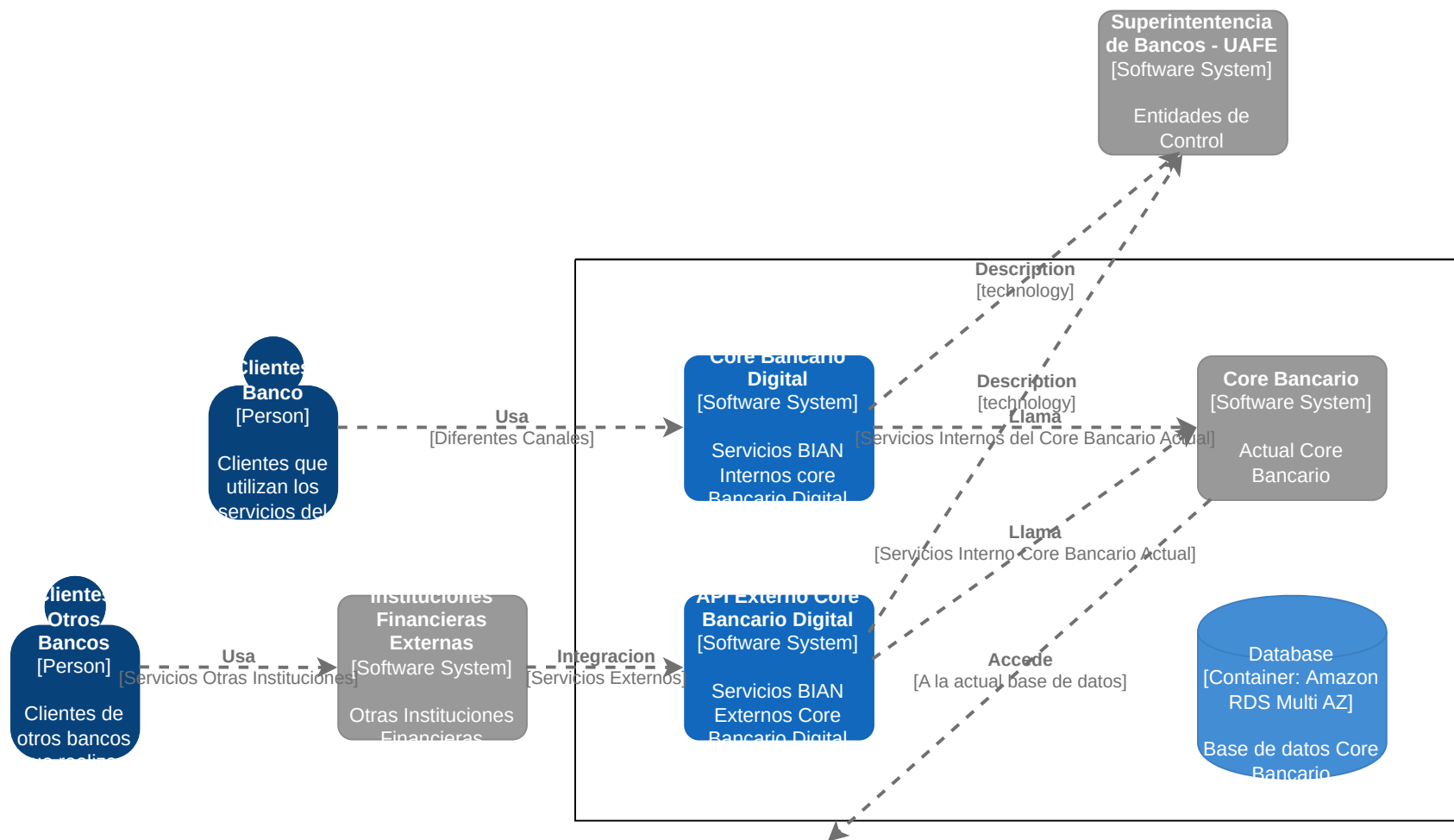
Para implementar el nuevo diseño, se debe comenzar con una migración inicial de la infraestructura. En este caso, se recomienda usar EC2 para la base de datos, y considerar la migración a una solución administrada por AWS, que aplicará actualizaciones automáticamente.

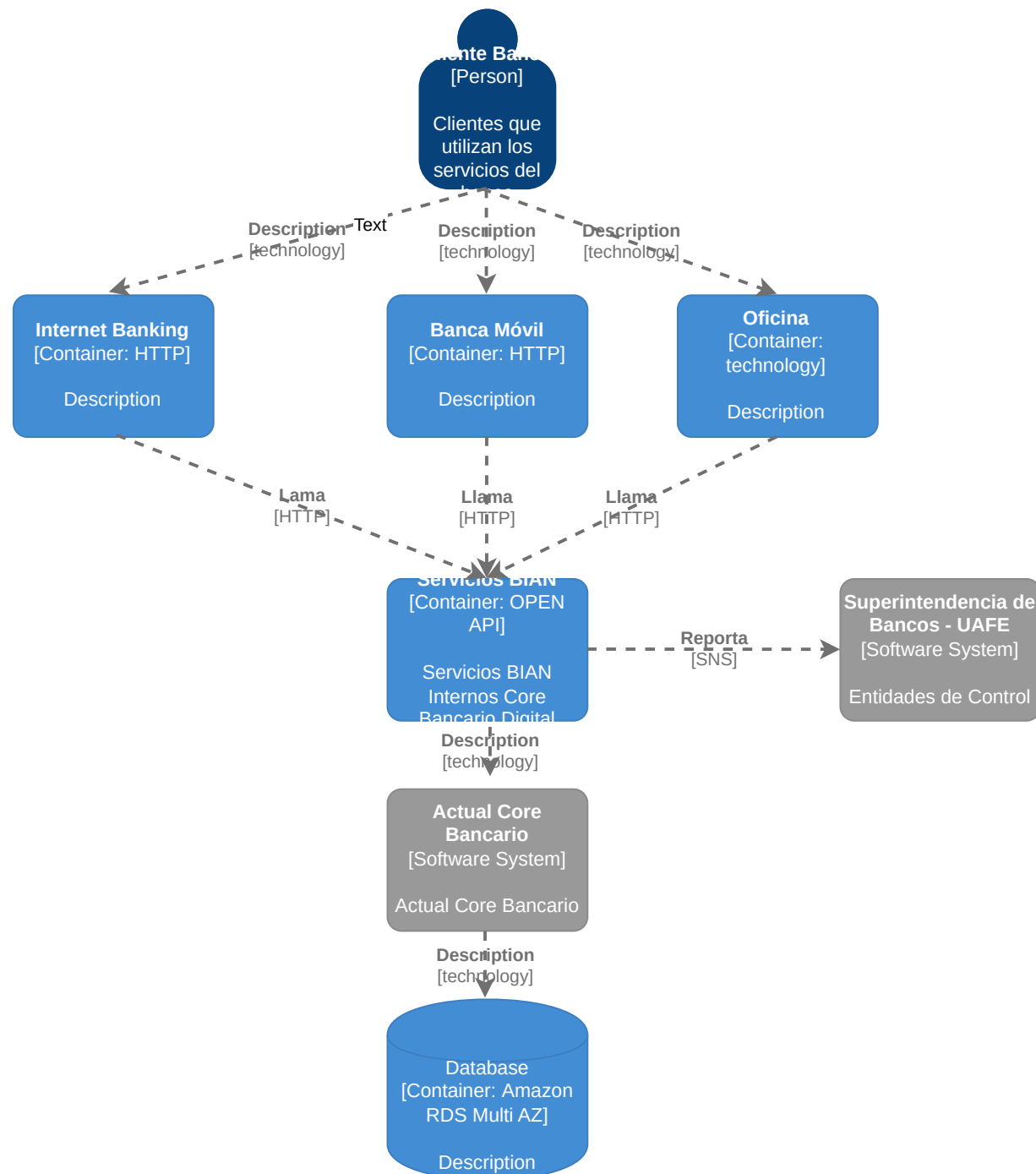
Inicialmente, los servicios API funcionarán como un intermediario con los servicios del núcleo actual. Con el tiempo, estos servicios deberían asumir el control progresivamente, lo que permitirá la eliminación de las instancias EC2 y la transición hacia microservicios independientes.

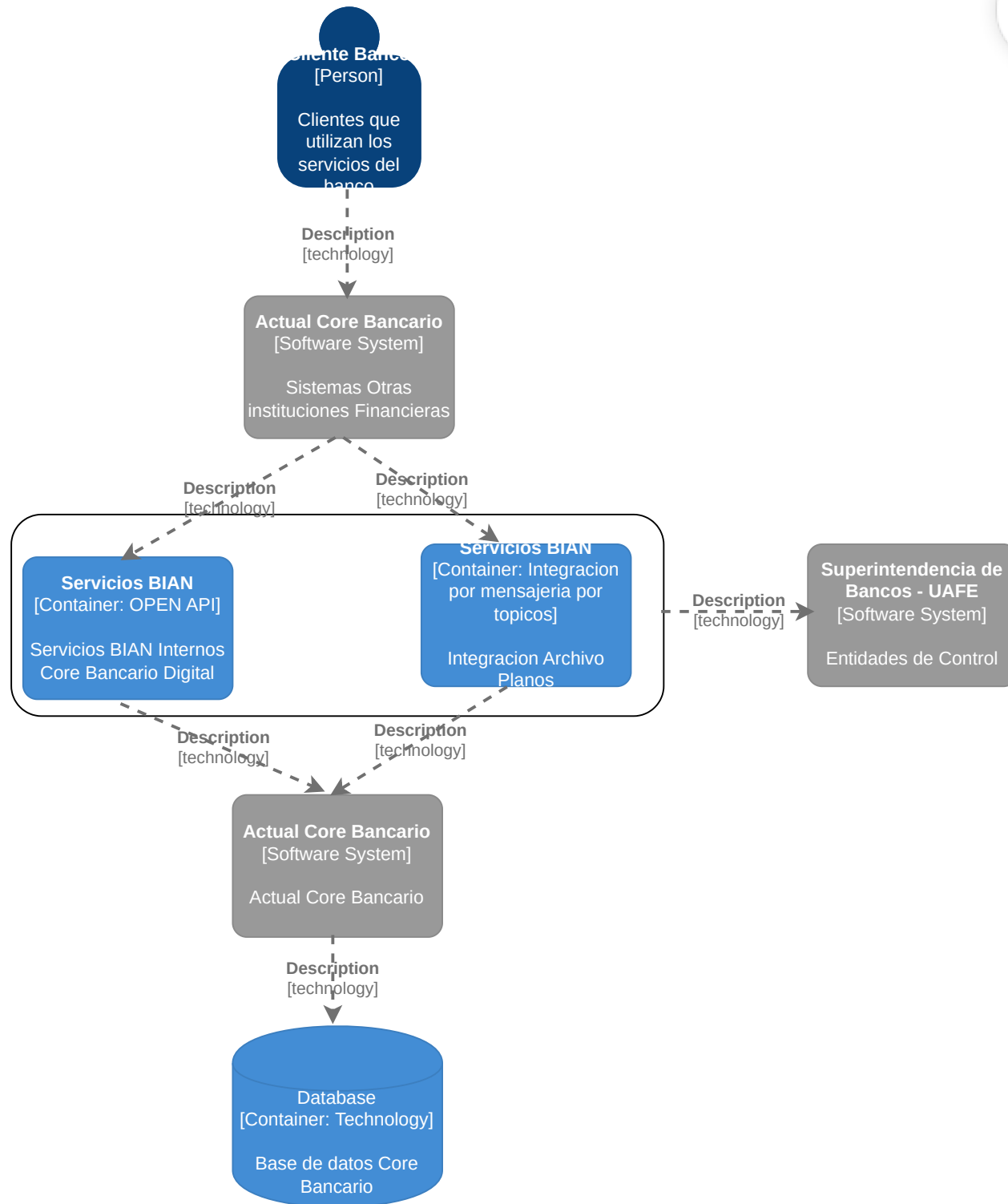
Se recomienda utilizar mensajería basada en tópicos para las integraciones con otros bancos, servicios de banca por Internet y aplicaciones móviles, con el fin de gestionar los acuerdos de nivel de servicio (SLA) de las API de manera más eficiente.

Para las aplicaciones internas, se sugiere usar conexiones remotas a través de API directamente, ya que su SLA es más flexible debido a que se trata de aplicaciones de intranet.

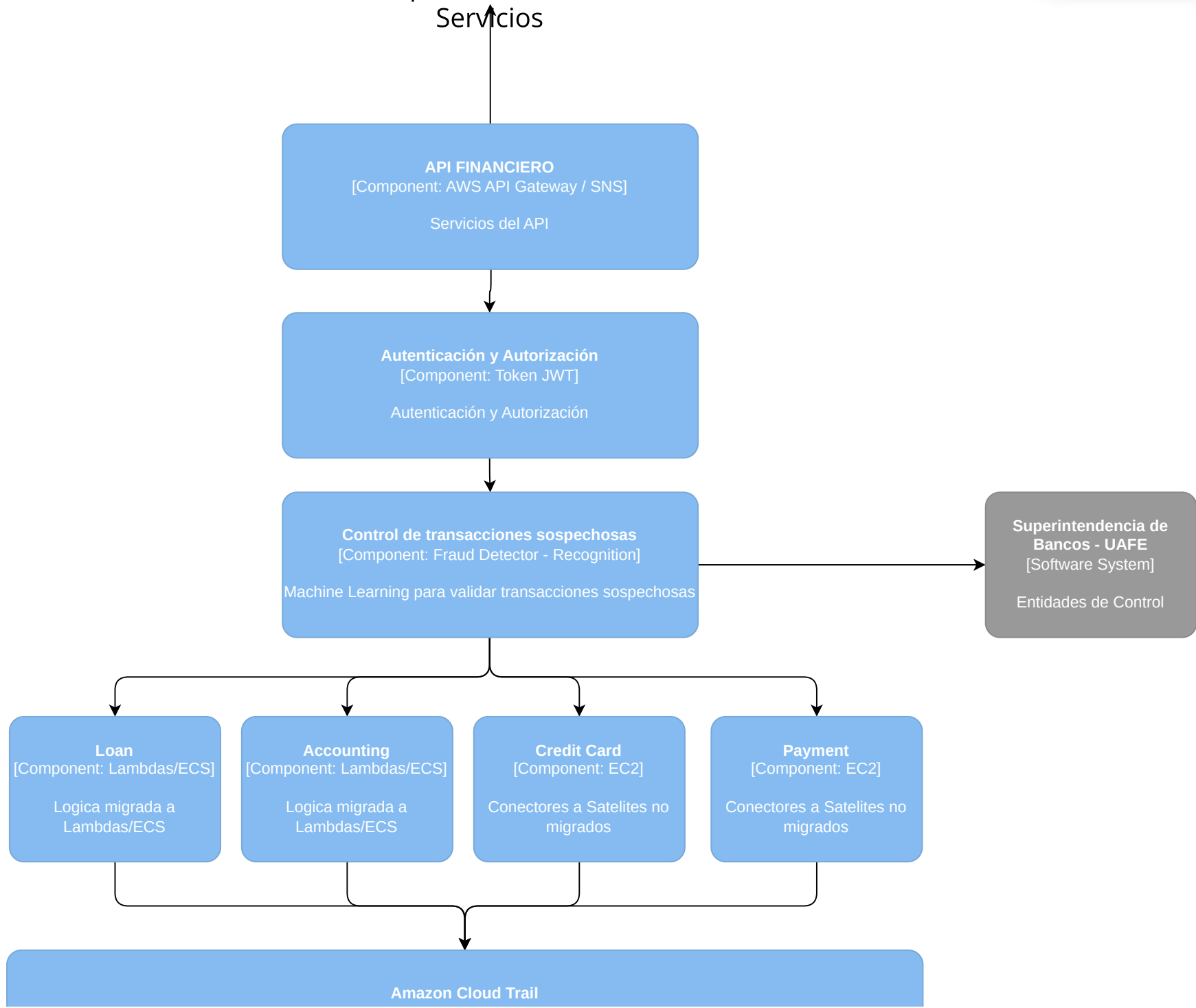








Arquitectura de los Servicios



[Component: Amazon Cloud Trail]

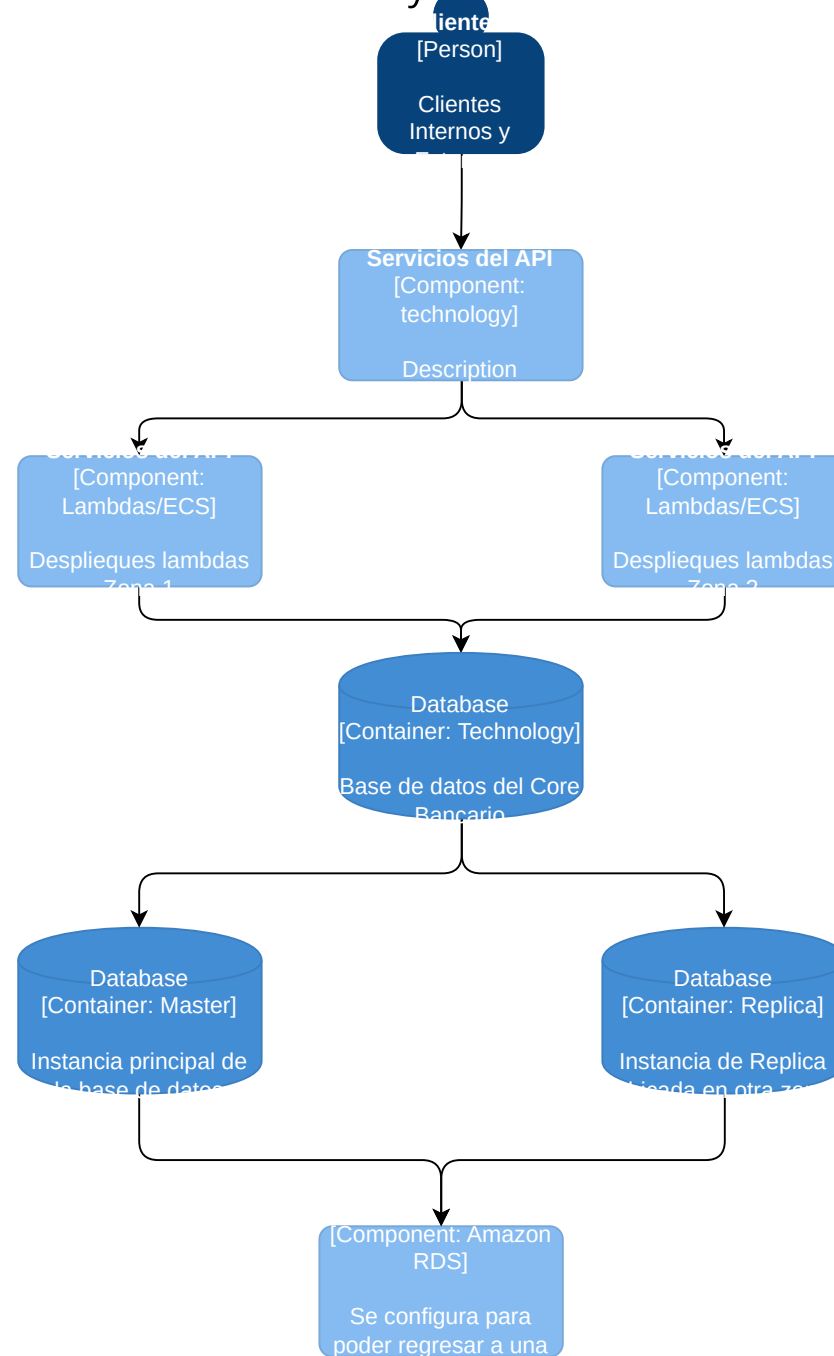
Auditoria del uso de las API

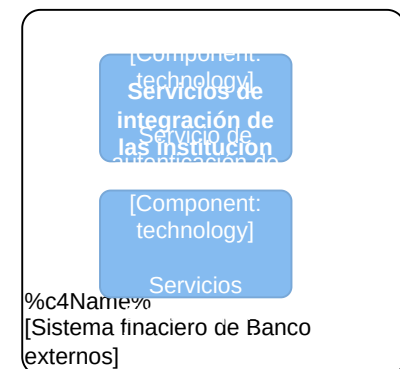
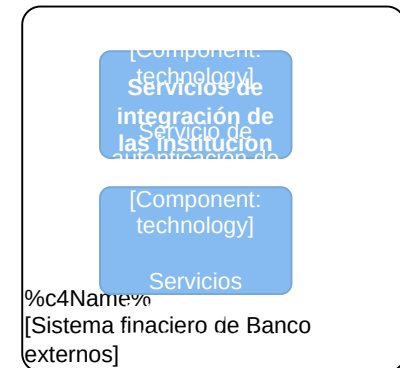
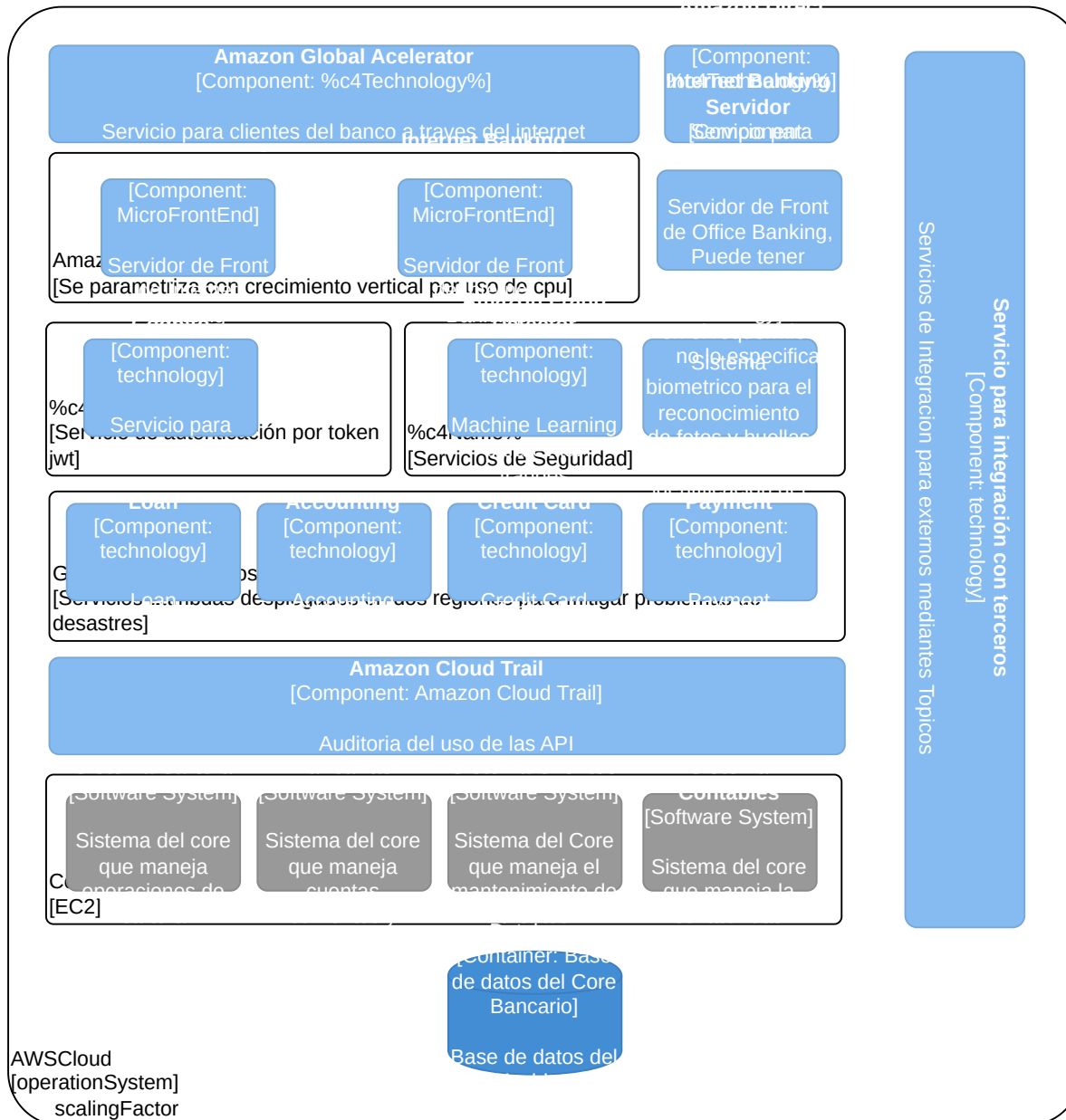


Database
[Container: Amazon RDS
Multi AZ]

Base de datos Core Bancario

Arquitectura de HA y RDS





o de
instituciones
[Person]

Sistemas Externos
[Software System]

Sistemas de otras Instituciones Financieras

Servicio para integracion con terceros
[Component: technology]

Servicios de Integracion para externos mediante Topicos

Cognito
[Component: technology]

%c4N% Servicio para autenticación jwt
[Servicio de autenticación por token jwt]

[Component: technology]

%c4N% Repositorio de integración para recibir archivos planos
[Servicio de autenticación por token jwt]

Loan
[Component: technology]

Go. Loan

Accounting
[Component: technology]

Accounting

Credit Card
[Component: technology]

Credit Card

Payment
[Component: technology]

Payment

[Servicios Lambdas desplegados en dos regiones para mitigar problemas de desastres]

Amazon Cloud Trail
[Component: Amazon Cloud Trail]

Auditoria del uso de las API

[Software System]

Cor. Sistema del core que maneja
[EC2]

[Software System]

Sistema del core que maneja

[Software System]

Sistema del Core que maneja

[Software System]

Sistema del core que maneja

Database

[Container: Base de datos del
Core Bancario]

Base de datos del core

AWSCloud
[operationSystem]
scalingFactor