

Gary Espinoza Sarzo
Ejercicio de Integración – DevSu

TAREAS

1. Diseñar una arquitectura de integración de alto nivel.

Especificado en los diagramas.

2. Especificar patrones de integración y tecnología a utilizar.

Especificado en los diagramas.

3. Abordar requisitos de seguridad, cumplimiento normativo, ley orgánica de protección de datos personales.

- Data en reposo cifrada con AES 256 utilizando el feature Transparent Data Encryption (TDE) de Azure SQL Database.

- Data sensible enmascarada con el feature Dynamic Data Masking de Azure SQL Database, esto con el fin de no dejar visible los datos sensible al momento de consultarlos (puede ser un enmascarado total o parcial).

- Data cifrada en tránsito con TLS 1.2

- Autenticación y Autorización con Microsoft Entra ID (OIDC) y Oauth 2, usando el grant flow de Authorization Code With PKCE para Mobile App y Web App.

4. Proponer estrategia para garantizar alta disponibilidad y recuperación ante desastres.

- Para la alta disponibilidad propongo un despliegue multirregión en modo activo-activo, para lo cual hago uso de Azure Front-Door para realizar el balanceo.

Adicionalmente hago uso de Azure API Management en modo global lo cual me permite poder desplegar una instancia de API GW en cada región, además hago uso de CosmosDB para que las DBs de las diferentes regiones puedan sincronizarse de manera automática haciendo uso del feature ChangeFeed.

5. Proponer estrategia de integracion multicore.

A medida que el nuevo Core se va desarrollando se puede ir migrando gradualmente el Core existente. Para lo cual una alternativa podría ser que 1ero consulte el nuevo Core y de no encontrar el registro ir al antiguo Core, otra alternativa podría ser usar la fecha para decidir a que Core ir, dependiendo del contexto de puedo plantear la mejor solución.

6. Delinear un enfoque para la gestión de identidad y acceso en todos los sistemas.

- Al autenticarse satisfactoriamente con Microsoft Entra ID este emitirá un token en formato JWT el cual contendrá claims que serán usados para validar los permisos.

Todos los componentes dentro de la arquitectura se comunicarán a través de una IP privada.

7. Propongo usar un nuevo API GW para exponer las APIs externas las cuales consumirán las APIS Core ya sea de forma independiente o haciendo orquestación.

8. Propongo usar una herramienta de API Management para registrar el inventario de APIs haciendo uso de Open API para documentar las APIs aplicando el patrón de contract-first para el desarrollo de las APIs.

9. Al hacer uso de microservicios (MS) se entiende que cada uno de estos es independiente, por lo cual se podría empezar por los MS menos críticos, luego los de nivel medio y por ultimo los critico. Por otro lado antes de realizar la migración es una buena practica poder usar Azure Site recovery para poder replicar toda la data que vamos a migrar.

El ejercicio se puede hacer mucho mas detallado, pero lastimosamente no cuento con tanto tiempo para hacerlo, en todo caso en una reunión podríamos entrar a mas detalle de cualquier punto.

Gracias.







